



TEKNILLINEN TIEDEKUNTA

KONENÄÖN HYÖDYNTÄMINEN TILAN VALVONNASSA

Eetu Karvonen

KONETEKNIikka

Diplomityö

Syyskuu 2021

TIIVISTELMÄ

Konenäön hyödyntäminen tilan valvonnassa

Eetu Karvonen

Oulun yliopisto, Konetekniikan tutkinto-ohjelma

Diplomityö 2021, 69 s. + 1 liite

Työn ohjaaja yliopistolla: Toni Liedes

Tämän työn tavoitteena on kehittää konenäköjärjestelmää, joka tunnistaa objekteja videosityötteestä. Konenäköjärjestelmän avulla pyritään parantamaan tilan valvontaa. Työn lähtökohtana on kehittää konenäköohjelmistoa, joka kykenee havaitsemaan, missä päin tilaa henkilöt sijaitsevat. Työhön liittyviä konenäön osa-alueita on objektien havaitseminen ja tunnistus. Työn edetessä tutkimuskohteeksi tarkentui uudelleentunnistus useamman reaaliaikaista videokuvaa tuottavan kameran välillä.

Työssä perehdyttiin aluksi konenäön, objektin tunnistuksen ja koneoppimisen teoriaan sekä tutustuttiin aiheeseen liittyviin aiempiin tutkimuksiin. Teoriakatsauksesta kertyneiden tietojen avulla lähdettiin kehittämään uudelleentunnistusohjelmistoa. Kehitystyössä käytettiin hyödyksi avoimeen lähdekoodiin perustuvia ohjelmistoja ja ohjelmistokirjastoja. Ohjelmiston taustalla on valmiiksi opetettuja tekoälymalleja, jotka vastaavat objektien tunnistamisesta ja seurannasta sekä uudelleentunnistuksesta. Uudelleentunnistusohjelmiston toimivuutta arvioitiin tekemällä kontrolloituja sekä todellisia olosuhteita jäljitteleviä testejä. Testien perusteella prototyyppiohjelmiston toimivuus oli hyvä. Ohjelmisto vaatii kuitenkin jatkokehittämistä, jotta ohjelmistosta saataisiin toimiva tuote.

Asiasanat: Tekoäly, konenäkö, sisätilapaikannus, uudelleentunnistus

ABSTRACT

Utilization of the computer vision in surveillance of the space

Eetu Karvonen

University of Oulu, Degree Programme of Mechanical Engineering

Master's thesis 2021, 69 pp. + 1 Appendix

Supervisor at the university: Toni Liedes

Purpose of this master's thesis is to develop a computer vision system that recognizes objects from the video feed. The goal is to utilize the computer vision system for surveillance of the space and to develop a system that can detect where are the people located in the space. Object detection and recognition are the sectors of the computer vision that this thesis focuses on. Subject of the thesis narrowed down to re-identification during the thesis.

First, the theory of computer vision, object recognition and machine learning was studied and previous studies regarding the subject was researched. Using the knowledge gained from the theory section, the developing of the re-identification program was started. In the development of the re-identification program open-source libraries and frameworks were used. On the background of the software were pre-trained deep learning models that were responsible of the object recognition, tracking and re-identification. The results of the re-identification program were verified using controlled and real-world mimicking tests. According to the tests the results of the prototype program were good. Although, the program requires further development to refine to finished product.

Keywords: Artificial intelligence, computer vision, indoor positioning, re-identification

ALKUSANAT

Tämä diplomityö on tehty kevään ja kesän 2021 aikana Oulussa. Isot kiitokset työnantajalleni mielenkiintoisesta aiheesta ja mahdollisuudesta työskennellä heidän kanssaan. Erityiskiitokset työkavereilleni Esalle ja Juhalle, jotka ohjasivat työn kulkua yrityksen puolelta. Suuri kiitos myös Tonille työn ohjaamisesta yliopiston puolesta sekä erittäin laadukkaasta opetuksesta vuosien varrella. Tonille kuuluu kiitos myös diplomityönantajan löytymisestä.

Oulu, 27.09.2021

Eetu Karvonen

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

ALKUSANAT

SISÄLLYSLUETTELO

MERKINNÄT JA LYHENTEET

1 JOHDANTO	8
1.1 Tausta ja tavoitteet	8
1.2 Työn rakenne.....	9
2 KONENÄKÖ	10
2.1 Kuvan muodostus.....	11
2.2 Esikäsittely	13
2.3 Kuvan segmentointi	14
2.3.1 Reunantunnistus	15
2.4 Kuvan luokittelu.....	16
2.5 Objektin havaitseminen ja tunnistus	16
2.5.1 Objektin havaitseminen.....	17
2.5.2 Objektin seuranta	18
2.6 Oppiminen ja neuroverkot.....	19
2.6.1 Neuroverkot.....	20
2.6.2 Neuroverkkojen arkkitehtuuri	21
2.6.3 Neuroverkkojen opettaminen	24
3 SISÄTILAPAIKANNUS JA UUELLEENTUNNISTUS	25
3.1 Visuaalinen sisätilapaikannus	25
3.2 Uudelleentunnistus	26
3.3 Lentokenttäympäristöön tehty uudelleentunnistuksen toteutus	27
3.3.1 Havainnointi, seuranta ja uudelleentunnistus.....	27
3.3.2 Toteutuksessa vastaan tulleet haasteet	28
3.3.3 Arkkitehtuuri	29
3.3.4 Datan kerääminen, käyttöliittymä ja uudelleentunnistustulokset.....	30
3.3.5 Suositukset	30

3.4 Torchreid-kirjasto uudelleentunnistukseen	31
4 TEKOÄLYN EETTINEN KEHITYS.....	32
5 TEORIAOSUUDEN YHTYMÄKOHDAT TYÖN KÄYTÄNTÖÖN	34
6 OBJEKTIN TUNNISTUS- JA SEURANTAOHJELMA.....	37
7 SOVELLUKSEN KEHITTÄMINEN.....	41
7.1 Kehitysympäristön käyttöönotto ja testaus	41
7.2 Uudelleentunnistuksen kehittäminen	42
8 TESTIT JA ANALYSOINTI.....	46
8.1 Ensimmäinen testaus	46
8.2 Sovelluksen parantaminen ja uusi testaus	48
8.3 Useamman kameran testaus	53
8.4 Tulosten yhteenveto ja arviointi.....	55
8.5 Jatkokehitys.....	56
8.6 Sovelluskohteiden pohdintaa	57
9 EETTISTEN KYSYMYSTEN HUOMIOIMINEN KÄYTÄNNÖSSÄ	60
10 YHTEENVETO	62
11 LÄHDELUETTELO.....	64

LIITEET:

Liite 1. Uudelleentunnistusohjelmiston white paper.

MERKINNÄT JA LYHENTEET

COCO	Common Objects in Context
DDS	Data Distribution Service
EU	Euroopan unioni
FPS	Frames per Second, kehystä sekunnissa
GPS	Global Positioning System, Maailmanlaajuinen paikannus järjestelmä
ID	Identification, yksilöllinen tunniste
MoGs	Mixtures of Gaussians, Gaussin sekoitemalli
REID	Re-identification, Uudelleentunnistus
RoI	Region of Interest, Mielenkiinnon alue
RTSP	Real-Time Streaming Protocol
b_k	Bias, neuronin k vakiotermi
P	Epävarmuustekijät
w_{kj}	Neuronien k ja j välinen painoarvo
x	Sisääntulo
y	Ulostulo

1 JOHDANTO

2020-luvulla tekoäly on hyvin laajalti yhteiskunnassamme läsnä, vaikka se onkin hieman piilossa. Tekoälyä on internetin hakukoneiden taustalla ja erilaisten sisällönsuosittelualgoritmien voidaan myös ajatella olevan tekoälyä. Tekoälyyn liittyy paljon erilaisia käsitteitä, kuten neuroverkot ja syväoppiminen.

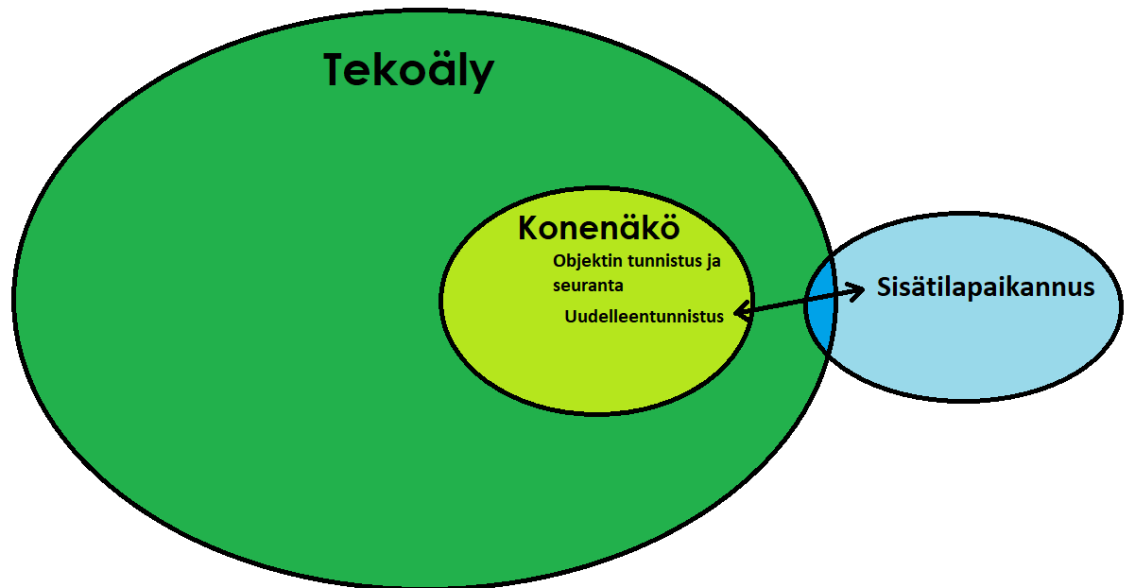
1.1 Tausta ja tavoitteet

Tässä työssä tarkoituksena on keskittyä yhteen tekoälyn osa-alueeseen, konenäköön, eli miten videokuvasta tunnistetaan tekoälyn avulla objekteja ja tapahtumia. Lisäksi tarkoituksena on tutkia ja parantaa sovelluskohteita, joita videokuvatunnistuksella voidaan tehdä. Konenäkö on yksi tekoälyn yleisimmistä käyttökohteista ja sitä voi käyttää hyvin monenlaisiin käyttötarpeisiin. Lääketieteessä konenäköä voi käyttää apuna tunnistamaan poikkeamia, kuten kasvaimia, elimistöstä. Teollisuudessa konenäköä voi käyttää esimerkiksi lajittelurobotin apuna tunnistamaan valmistusvirheitä linjastolta ja lajittelemaan virheelliset tuotteet robotin avulla. Tässä työssä konenäköä käytetään tilan valvontaan.

Tarkoituksena on käyttää videokuvatunnistusta sisätilapaikannukseen. Sovelluskohde voi olla esimerkiksi teollisuushalli, josta saadaan videokuvaa. Sisätilapaikannuksen lisäksi pohditaan muita käyttökohteita, kuten voidaanko tekoälyn avulla videokuvasta havaita turvallisuuspoikkeamia ja antaa poikkeamista hälytys tai muu vastaava ilmoitus. Turvallisuuspoikkeamat voivat olla esimerkiksi havaintoja alueelle kuulumattomista esineistä tai henkilöistä.

Signaaleihin (WiFi, Bluetooth, ZigBee...) perustuvia sisätilapaikannusmenetelmiä on aikaisemmin tutkittu laajalti, mutta tässä työssä tutkitaan videokameroihin perustuvaa menetelmää. Signaaleihin perustuvissa menetelmissä on omat haasteensa, kuten signaalien aiheuttamat häiriöt sekä laiteinfrastruktuuri (Calderoni ja muut 2014, s. 1; Wei & Akinci 2019, s. 3). Tässä työssä on haluttu tutkia nimenomaan kameroihin perustuvaa sisätilapaikannusta. Samalla päästään tutkimaan myös muita konenäköön liittyviä aiheita ja tekoälyä.

Konenäkö pitää sisällään paljon erilaisia tehtäviä, kuten kuvien luokittelua ja objektien havaitsemista. Tässä työssä keskitytään erityisesti objektin tunnistukseen ja seurantaan sekä uudelleentunnistukseen. Uudelleentunnistuksella tarkoitetaan prosessia, jossa tunnistetaan, onko henkilö havaittu kameraverkostossa jo aikaisemmin (IGI Global 2021). Uudelleentunnistusta pyritään käyttämään hyödyksi sisätilapaikannukseen. Kuvassa 1 on kuvattu työhön liittyvää käsitteistöä.



Kuva 1. Kaavio työhön liittyvistä käsitteistä.

1.2 Työn rakenne

Työn rakenne koostuu teoriaosuudesta ja käytännön osuudesta. Teoriaosuudessa (kappaleet 2–4) on perehdytty aihealueeseen. Kappaleessa viisi on pohdittu teoriaosuudesta saatujen tietojen yhtymäkohtia tämän työn käytännön tekemiseen. Kappaleissa 6–8 teoriaosuudesta saatuja tietoja on hyödynnetty käytännössä uudelleentunnistusohjelmiston kehittämiseen sekä testaamiseen. Kappaleessa yhdeksän on kirjoittajan omaa pohdintaa sovelluksen eettisyyteen liittyen. Lopuksi työn yhteenveto on kappaleessa kymmenen.

2 KONENÄKÖ

Tekoäly on viime aikoina ollut kovasti pinnalla yhtenä suurena tulevaisuuden suuntauksena. Tekoälyn on pohdittu olevan apuna monella eri alueella, kuten lääketieteessä, kauppatieteessä, lakimiesten päätöksenteon tukena sekä tietenkin tekniikan alueella.

Tekoälyn historian kerrotaan ulottuvan jopa antiikin aikoihin asti. Jo silloin, kirjallisuudessa on ollut ajatus tekoälyllisistä olennoista, jotka voisivat toimia älykkäästi käsityöläisinä. Tekoälyn modernimman historian voidaan katsoa alkaneen kuitenkin vasta 1900-luvulla, kun ensimmäiset tietokoneet tekivät tuloaan. Tekoälytutkimuksen katsotaan varsinaisesti alkaneen 1956 Dartmouthin konferenssissa. Kyseessä oli noin kahdeksan viikon mittainen työpaja, jonka osallistujat olivat seuraavien vuosikymmenien ajan tekoälytutkimuksen suurimpia tutkijoita. Tietotekniikan sekä tekoälyn huima kehitys mahdollisti asioita, jotka olivat vielä vähän aikaa sitten tuntuneet mahdottomilta suorittaa koneellisesti. Useat Dartmouthin työryhmän jäsenet ennustivat, että koneista voitaisiin kehittää ihmistä älykkäämpiä yhden sukupolven aikana. Suuret lupaukset toivat valtavasti rahoitusta, mutta pian selvisikin, että tekoälyn haasteet oli aliarvioitu. Tämä johti rahoituksen vähenemiseen ja niin kutsuttuun tekoälytalveen. Tämän jälkeen tekoäly on välillä nostanut enemmän päätään ja välillä taas hiipunut uudelleen. Yleinen tietotekniikan kehitys on mahdollistanut uusia tekoälyaaltoja. Tekoälyn nousukausiin on myötävaikuttanut monesti yksittäiset saavutukset, kuten tekoälyn kehittyminen ihmistä paremmaksi esimerkiksi shakissa ja go-lautapelissä. (Tekoäly.info, 2021)

Konenäkö on yksi tekoälyn osa-alue, joka keskittyy tekoälyn hyödyntämiseen visuaalisen materiaalin, kuten kuvien ja videoiden, analysoinnissa. Konenäössä keskeisessä osassa on kuvan muodostus ja tietokoneelle kaappaaminen. Tämän jälkeen tapahtuu varsinainen tekoälyn hyödyntäminen. Se, mitä konenäön ja tekoälyn hyödyntäminen käytännössä on, riippuu sovelluskohteesta ja siitä mitä konenäöllä halutaan saada aikaan. Yleisimpiä konenäkösovelluskohteita on erilaiset objektien tunnistamistehtävät.

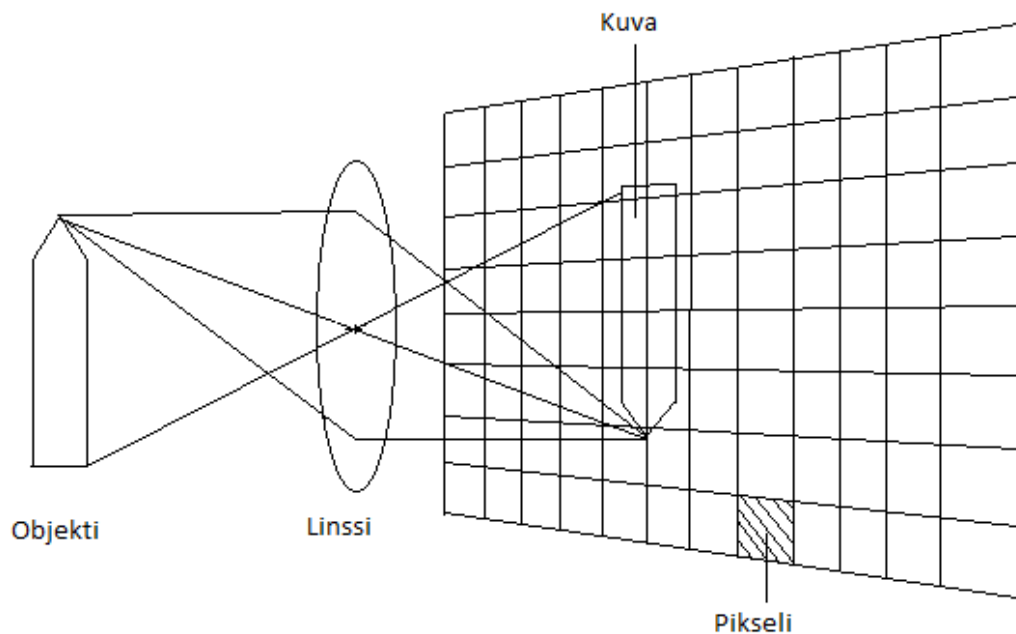
2.1 Kuvan muodostus

Konenäköprosessi alkaa kuvan muodostamisesta. Vielä jokin aika sitten konenäköä käsittelevissä teksteissä kuvan muodostus on ollut suuressa roolissa. Kuvan muodostuksella on oleellinen rooli siihen, miten hyvin konenäköä voidaan hyödyntää. Nykyään laajasti ja halvalla saatavilla olevat kamerat ovat mahdollistaneet sen, että laadukastakin kuva- ja videokuvamateriaalia on saatavilla runsaasti. Tästä aiheutuu se, ettei kuvan muodostuksella ole enää niin suurta roolia, vaan konenäköä pitäisi pystyä soveltamaan mihin tahansa saatavilla olevaan kuvamateriaaliin. Aikaisemmissa konenäköä käsittelevissä teksteissä esiin nousee erilliset konenäköä varten suunnitellut kamerat, mutta nykyään kuluttajamarkkinoiden kamerat ovat usein riittävän hyviä konenäkösovellusten käyttöön. Lisäksi konenäkösovellusten käyttökohteet ovat monesti sellaisia, että konenäköä halutaan käyttää muutenkin yleisesti saatavilla olevaan kuvamateriaaliin, sen sijaan, että haluttaisiin käyttää erikseen konenäkökameralla kuvattua materiaalia. Toki konenäköä varten tehty erilliset kamerat sisältävät monipuolisempia ominaisuuksia ja ovatkin spesifeihin käyttötarkoituksiin monesti paljon parempi vaihtoehto kuin yleiskäyttöinen kamera. Nykyään on kuitenkin yleistä, että aivan yleiskäyttöisetkin kamerat sisältävät verrattain korkeatasoisia konenäkö ominaisuuksia, kuten kasvojen havaitsemista.

Kuvan muodostaminen koostuu kameran, siinä olevien linssien sekä valaistuksen yhteistoiminnasta. Linssi heijastaa valonsäteet kameran kuvasensorille, joka reagoi vastaanottamansa valon intensiteetin mukaan muuttaen valon energiaa sähköiseksi varaukseksi. Kuvasensori on jaettu useampaan pienempään osaan, pikseliin. Pikseleitä voi olla sensorissa esimerkiksi 1920 kappaletta sivusuunnassa ja 1080 kappaletta pystysuunnassa, eli yhteensä yli kaksi miljoonaa pikseliä. Kuvassa 2 on havainnollistettu kuvan muodostamista yksinkertaistettuna 12 x 8 pikselin sensorilla. Yksittäiset pikselit aistivat valon intensiteettiä ja muuttavat tiedon sähköiseksi varaukseksi. A/D muuntamalla jokaisen pikselin muodostama informaatio saadaan muodostettua kuvan bittikartta. Bittikartta voidaan esittää kaksiulotteisena matriisina, jonka alkioina on yksittäisten pikseleiden valointensiteettiarvot. Binäärikuva on yksinkertaisin mahdollinen kuva, jossa bittikartan alkiot ovat joko 0 tai 1, tarkoittaen yleisimmin värejä musta tai valkoinen. Seuraavaksi yksinkertaisin kuva on harmaasävykuva, jossa bittikartan alkiot saavat arvon 0–255. Sopimuksesta riippuen 0 voi tarkoittaa esimerkiksi täysin mustaa ja 255 täysin valkoista pikseliä

ja jokainen arvo 0:n ja 255:n väliltä jotain harmaan sävyä muuttuen lineaarisesti vaaleammaksi arvon kasvaessa.

Kolmas ja yleisimmin käytetty kuva on värikuva. Siinä kukin pikseli sisältää esimerkiksi kolme väriarvoa, yhden punaisen, yhden vihreän ja yhden sinisen valon intensiteetin. Yleensä intensiteettiä kuvataan 8-bittisellä luvulla, joka mahdollistaa 256 arvoa eli arvot 0–255. 256 arvoa mahdollistaa yleensä riittävän tarkan tarkkuuden, mutta toki on mahdollista käyttää esimerkiksi 10-bittistä enkoodausta, jolla saadaan 1024 mahdollista intensiteetin arvoa, joka lähestyy jo ihmissilmän havainnointikyvyn rajoja. (Shapiro & Stockman 2001, s. 29)



Kuva 2. Kameran kuvasensorin toiminta yksinkertaistettuna 12 x 8 pikselin sensorilla.

Digitaalinen videokuva koostuu sarjasta yksittäisiä kuvia. Videokuvassa yksittäiset kuvat, eli niin sanotut kehykset, vaihtuvat käyttökohteesta riippuen tyypillisesti 15–60 kertaa sekunnissa. Kuvan muodostamisen jälkeen kuva täytyy saada kaapattua tietokoneelle, jolla tekoälytoiminnot tapahtuvat.

2.2 Esikäsittely

Kun kuva on saatu muodostettua ja kaapattua tietokoneen muistiin, on seuraava askel perinteisesti kuvan esikäsittely. Tosin jotkin älykamerat voivat tehdä jonkinlaista esikäsittelyä myös ennen varsinaista kuvan kaappaamista tietokoneelle. Kuten kuvanmuodostamisen merkitys, myös esikäsittelyn merkitys on pienentynyt viime aikoina parantuneiden kameroiden johdosta. Esikäsittelyn tarkoituksena on parantaa kuvan laatua joko ihmissilmää varten tai konenäön tapauksessa monesti parantamaan itse konenäköprosessia. Raja esikäsittelyn ja varsinaisen käsittelyn termien välillä voi olla häilyvä, ja tässä tekstissä esikäsittelyllä tarkoitetaan kaikkea käsittelyä, joka tapahtuu ennen varsinaisia tekoäly toimintoja. Esikäsittelytoimintoja voi olla esimerkiksi häiriötekijöiden pois suodattaminen, kuvan kääntäminen, valotuksen ja/tai kontrastin parantaminen sekä pienempien osa-alueiden irrottaminen. Käytettävät esikäsittelymenetelmät riippuvat käytettävästä konenäkösovelluskohteesta ja siitä mitä konenäöllä pyritään saamaan aikaan. (Hyyti 2016, s.14; Shapiro & Stockman 2001, s. 128)

Konenäköön liittyvät ongelmat voidaan jakaa karkeasti kolmeen yleisimmin esillä olevaan osa-alueeseen: kuvan segmentointiin, kuvan luokitteluun sekä objektin havaitsemiseen ja tunnistamiseen kuvasta. Raja näiden kolmen osa-alueen sekä myös esikäsittelyn välillä on häilyvä.

2.3 Kuvan segmentointi

Kuvan segmentoinnilla tarkoitetaan kuvan jakamista merkityksellisiin osiin. Kuvassa 3 on havainnollistettu segmentointia.



Kuva 3. Segmentointi havainnollistettuna. (Julkaistu Creative Commons -linsessillä. COCO Consortium 2020)

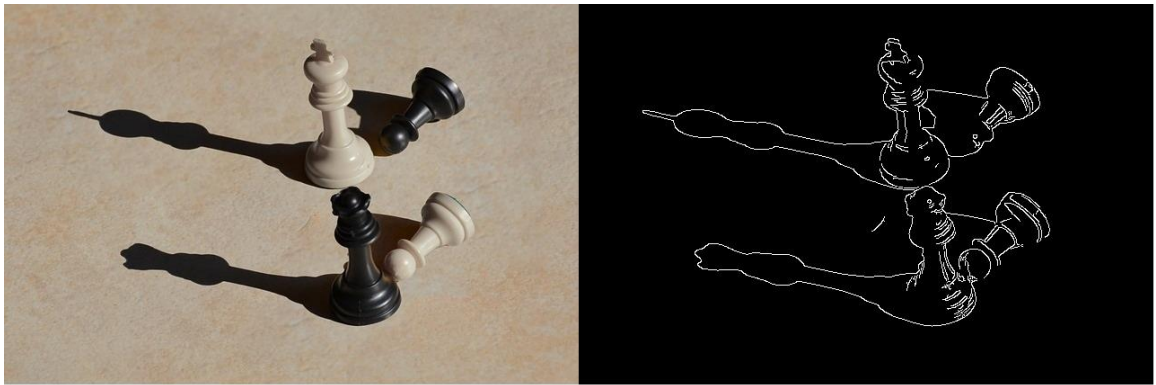
Segmentointi on juuri aihe, joka ei ole selkeästi rajattu omaksi osa-alueekseen. Segmentoinnin voi hyvin monessa tapauksessa ajatella kuuluvan osaksi esikäsittelyä. Lisäksi segmentointi liittyy monesti kuvan luokitteluun, (mikäli segmentoidut alueet halutaan luokitella), joka on itsessään jo toisenlainen ongelma.

Segmentointia on mahdollista tehdä monella eri tavalla. Kuvan 3 esimerkissä segmentointi on tehty erottaman eri objektit omiksi alueiksi, mutta segmentointi voidaan tehdä myös esimerkiksi värien perusteella, jolloin alueet muodostuvat lähes saman värisistä alueista. Tällöin esimerkiksi henkilö ei ole yksi kokonainen alue, vaan ruskea paita on yksi alue, ja eriväriset housut ovat toinen alue. Käytettävä segmentointimenetelmä riippuu käyttökohteesta ja tarvittavasta informaatiosta, mitä segmentoinnilla pyritään saamaan aikaan.

2.3.1 Reunantunnistus

Eräs tärkeä ja yleinen segmentointiin liittyvä menetelmä on reunojen ja ääriviivojen tunnistus (*eng. edge detection*). Reunantunnistus on yksi ensimmäisistä varsinaisista tekoälymenetelmistä, jota käytetään kuvan käsittelyn aikana. Reunantunnistuksen voi ajatella myös kuuluvan esikäsittelyyn, joten jälleen raja esikäsittelyn ja varsinaisen käsittelyn välillä on häilyvä. Yleensä ennen reunantunnistusta kuitenkin suodatetaan häiriöt pois, sillä ne voivat haitata reunantunnistusprosessia.

Reunantunnistus on kohtalaisen yksinkertainen tapa saada tiettyihin tilanteisiin relevanttia tietoa. Reunantunnistuksen avulla informaatioarvo kasvaa, mutta datan määrä pienenee huomattavasti. Reunantunnistus kannattaakin tehdä mahdollisimman aikaisessa vaiheessa, jotta mukana kuljetettavan datan määrä saadaan pieneksi mahdollisimman aikaisessa vaiheessa. Kuvassa 4 on havainnollistettu reunantunnistusta ja sitä, miten huomattavasti vähemmän dataa kuvassa on reunantunnistuksen jälkeen.



Kuva 4. Reunojentunnistus tehty käyttäen OpenCV-nimisen Python-ohjelmakirjaston ”Canny” reunantunnistusalgoritmia. (Julkaistu Pixabay-linsessillä. Pixabay 2021)

Reunantunnistus etsii kuvasta rajat, joissa pikseleiden kirkkaus/väriarvot muuttuvat merkittävästi. Tällaisia rajakohtia voivat olla syvyysepäjatkuvuus (objektin fyysiset reunat), objektin pinnan suunnan epäjatkuvuus, pinnan heijastuksen epäjatkuvuus tai valoisuuden epäjatkuvuus (varjot). Kuvassa 4 on kaikkia neljän tyyppisiä rajakohtia.

Reunantunnistus ei yleensä ole täydellinen. Reunantunnistus ei välttämättä tunnista kaikkia rajapikseleitä sekä tunnistaa ”kohinareunoja”, jotka eivät oikeasti ole reunoja.

2.4 Kuvan luokittelu

Kuvan luokittelu liittyy hyvin vahvasti sekä segmentointiin, että myös objektin havaitsemiseen ja tunnistukseen. Kuten edellisessä kappaleessa mainittiin, segmentoidut alueet voidaan luokitella, jolloin alueet saavat ihmiselle ymmärrettävän luokan, kuten ”tie” tai ”rakennus”.

”Kuvan luokittelu” omana ongelmanaan tarkoittaa esimerkiksi tilannetta, jossa tekoälyn tehtävänä on luokitella, onko kuvassa kissa vai koira. Tämä muistuttaa objektin tunnistusta, mutta eroaa siten, että kuvan luokittelua varten tekoälylle annetaan vain luokiteltava piirre. Piirre voi olla esimerkiksi käsin kirjoitettu kirjain. Tekoäly tulkitsee, mikä kirjain on kyseessä, ja antaa kirjaimelle sitä vastaavan luokan.

2.5 Objektin havaitseminen ja tunnistus

Objektin havaitseminen ja tunnistaminen puolestaan on huomattavasti monipuolisempi ongelma kuin pelkkä kuvan luokittelu. Jotta kuvasta voidaan tunnistaa, mitä objekteja kuvassa on, objektit täytyy ensin havaita ja paikallistaa kuvaan. Ihmisille havainnollistettuna (kuva 5) havaitseminen voidaan ilmaista kuvaan piirrettävällä havaitun objektin ympäröivällä kehikolla (*eng. bounding box*).



Kuva 5. Kuvasta havaittu autot ja merkitty ympäröivillä kehikolla.

2.5.1 Objektin havaitseminen

Objektien havaitseminen perustuu objektien ominaispiirteisiin. Jotkin objektit ovat vaikeampia havaita kuin toiset. Esimerkiksi rakennusten ominaispiirteet vaihtelevat suuresti. Rakennuksia on hyvin monen kokoisia, muotoisia ja värisiä. Toisaalta taas kasvojen ominaispiirteet ovat universaaleja. Kasvot koostuvat vaaleammasta alueesta, jossa on tummemmat silmät sekä vertikaalit viirut kulmakarvojen ja suun alueella. Jalankulkijan tunnistamisessa ominaispiirteitä ovat torso, pää, kädet ja jalat. Sen lisäksi, että tunnistettaisiin pelkästään näitä ominaispiirteitä, jalankulkijan tunnistamisessa voidaan käyttää hyväksi myös ominaispiirteiden sijoittumista muihin ominaispiirteisiin nähden. Esimerkiksi pää ei ole koskaan yhteydessä jalkaan, joten mikäli tällainen tapaus kuvasta tunnistettaisiin, voitaisiin suurella todennäköisyydellä sanoa, että kyseessä on väärä tunnistus, eikä kyseessä oikeasti ole jalankulkija. (Russel & Norvig 2016, s. 943)

Eräs yleinen menetelmä objektin havaitsemiseen liittyen on niin kutsuttu liukuvan ikkunan menetelmä (eng. *sliding window*). Menetelmässä tietyn kokoinen ”ikkuna” tai

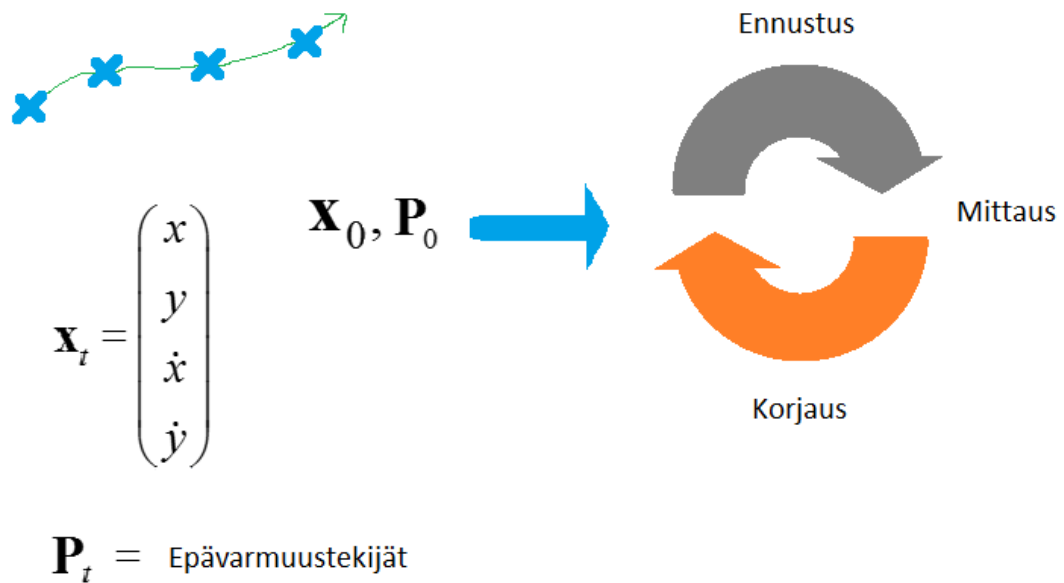
”kehys” käy koko kuvan läpi iteroiden liikuttamalla kehystä esimerkiksi pikselin verran aina eteenpäin. Kukin kehys vuorollaan tunnistaa, onko kehyksen sisällä tunnistettavissa objektia. Yleensä saman objektin tunnistaa useat lähekkäiset kehykset, joten ylimääräiset tunnistukset täytyy suodattaa jollain tavalla pois. Jotta kuvasta voidaan tunnistaa useamman kokoisia objekteja, täytyy liukuvan ikkunan menetelmä toistaa kuvan isompaan tai pienempään kokoon skaalatulle kovalle tai vaihtoehtoisesti käyttää useampaa eri kokoista kehystä. (Russel & Norvig 2016, s. 943)

2.5.2 Objektin seuranta

Objektin havaitsemiseen ja tunnistamiseen liittyy hyvin vahvasti myös objektin seuranta. Objektin seuranta tarkoittaa objektin seuraamista videokuvan yksittäisten peräkkäisten kuvien välillä. Vaikka periaatteessa on mahdollista jättää objektin seuranta kokonaan pois ja käyttää vain objektin tunnistusta jokaiseen videokuvan yksittäiseen kuvaan, on erikseen tehtävä objektin seuranta huomattavasti parempi vaihtoehto. Ilman objektin seurantaa tunnistuksesta tulee helposti epäjatkuva. Lisäksi objektilla ei ole yhteyttä kahden perättäisen kuvan välillä. Tämä voi vaikuttaa esimerkiksi videolla olevien objektien laskeamiseen. Myös jokaiselle kovalle erikseen tehtävä tunnistus on laskentatehon haaskausta, sillä objektin seurannalla voidaan hyödyntää edellisistä tunnistuksista saatua dataa.

Kalman-suodatin on eräs yleisimmistä objektin seurannassa käytetyistä menetelmistä. Kalman-suodatin ennakoii dynaamisen järjestelmän tilaa edellisten mittausten perusteella, ottaen huomioon kohinan tuoman epävarmuuden. Esimerkiksi objektin seurannassa tilamuuttujina voi olla objektin x ja y koordinaatit sekä nopeudet x' ja y' . Kalman-suodatin ennustaa seuraavan tilan sekä sen epävarmuuden. Tämän jälkeen tehdään todellinen mitta objektin sijainnista, (joka voi siis sisältää kohinaa), ja päivitetään Kalman-suodattimen tila sekä epävarmuustekijät. Kalman-suodattimen toimintaa on havainnollistettu kuvassa 6. (Jurić 2015)

Objektin seurannassa Kalman-suodattimen tehtävänä on siis luoda havaituille objekteille ”rata”, jota pitkin objekti kulkee. Yhteys uusien objektien havaintojen sekä ratojen välille tehdään esimerkiksi unkarilaisella algoritmilla.



Kuva 6. Kalman-suodattimen toimintaperiaate.

2.6 Oppiminen ja neuroverkot

Koneoppiminen on eräs keskeisistä käsitteistä konenäössä ja erityisesti objektin tunnistuksessa. Koneoppimisella tarkoitetaan kykyä parantaa suorituskkyä tulevilla tehtävissä tekemällä havaintoja ympäristöstä, eli siis oppimalla. Oppiminen voidaan jakaa kolmeen kategoriaan sen perusteella, minkälaisista palautteista oppimisprosessissa käytetään. Ohjaamaton oppiminen tarkoittaa oppimista, jossa oppiminen tapahtuu ilman, että annetaan varsinaista palautetta. Esimerkiksi kone voidaan opettaa ryhmittelemään asioita ilman, että sille annetaan mitään valmiita ryhmiä joihin ryhmitellä. Tällöin kone ryhmittelee asioita parhaaksi katsomallaan tavalla. Ohjattu oppiminen puolestaan tarkoittaa oppimista, jossa opettaminen tapahtuu syötteistä, joille on annettu oikeat ulostulot. Esimerkiksi tekoälylle voidaan antaa syötteinä kuvia, joiden ulostulona on tieto siitä mitä kuva esittää. Näin kone voidaan opettaa tunnistamaan erilaisia kuvia. Vahvistusoppiminen taas tarkoittaa oppimista, jossa tekoälylle annetaan palautteena, suoriutuiko tekoäly hyvin vai huonosti. (Russel & Norvig 2016, s. 693–695)

Ohjattu oppiminen on yleisin oppimisen muoto. Siinä tekoälylle annetaan sisääntulo-
ulostulo pareja

$$(x_1, y_1), (x_2, y_2), \dots (x_n, y_n),$$

missä x_i on sisääntulo ja y_i on ulostulo. Oppimisen tarkoituksena on löytää sellainen funktio, joka toteuttaa mahdollisimman hyvin $y = h(x)$ jokaiselle sisääntulo-ulostulo parille. Tätä funktiota kutsutaan *hypoteesiksi*. Yleensä opetusdata sisältää todella suuren määrän sisääntulo-ulostulo pareja. Kun hypoteesi onnistutaan löytämään, voidaan hypoteesia käyttää mihin tahansa myös opetusdatan ulkopuoliseen sisääntuloon, jolloin hypoteesi ennustaa ulostulon.

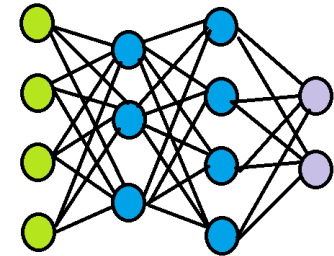
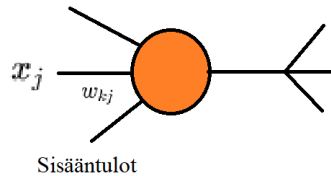
Yleensä täydellistä funktiota, joka ennustaisi ulostulon täysin oikein jokaisesta sisääntulosta, ei ole mahdollista löytää. Eräs mittari, jolla hypoteesin hyvyyttä voidaan mitata, on väärin ennustettujen ulostulujen osuus. Yleensä virheprosentti pienenee, kun opetusdatan kokoa kasvatetaan.

2.6.1 Neuroverkot

Erilaisia oppimismetodeita on monenlaisia. Eräs viime aikoina erityisesti pinnalla ollut menetelmä on neuroverkot. Neuroverkoilla tarkoitetaan arkkitehtuuria, joka on saanut inspiraationsa biologisten aivojen jäljittelemisestä. Moderni neuroverkkojen tutkimus onkin jakautunut kahteen leiriin. Toisessa pyritään mallintamaan todellisten neuronien sekä neuronijoukkojen toimintaa ja empiirisiä ominaisuuksia. Toinen puoli taas on kiinnostunut luomaan tehokkaita verkkoarkkitehtuureita ja -algoritmeja sekä ymmärtämään niiden matemaattisia ominaisuuksia. Neuroverkkoja on tutkittu jo tekoälyn alkuaajoista lähtien, mutta viime aikoina ne ovat nousseet erityisesti pinnalle. Tämän on mahdollistanut tietotekniikan yleinen kehitys sekä erityisesti 2010-luvulla lisääntynyt suurten datamassojen synty. Datamassoja käytetään hyödyksi neuroverkkojen opettamisprosessissa. (Russel & Norvig 2016, s. 25–28)

Neuroverkot koostuvat yksittäisistä neuroneista, jotka ovat kytkettyinä toisiinsa. Kuvassa 7 on kuvattu yksittäinen neuroni sekä neuroneista koostuva neuroverkko. Neuronien väleillä on omat painoarvonsa w_{kj} (*eng. weight*), jolla sisääntulo kerrotaan. Neuroni summaa jokaisen painotetun sisääntulon yhteen ja lisää summaan vakiotermin b_k (*eng. bias*).

Aktivaatiofunktion g tehtävä on laskea ulostulosta epälineaarinen kuvaus sekä ulostulon rajaaminen tietylle välille. Eräs käytetty aktivaatiofunktio on sigmoidi funktio, jolla ulostulo voidaan rajata välille $[0, 1]$. (AI Optio 2018)



Syötekerros Piilokerrokset Tuloskerros

$$y_k = g\left(\sum_{j=1}^m (w_{kj} x_j) + b_k\right)$$

Kuva 7. Yksittäinen neuroni sekä täysin kytketty, neljäkerroksinen neuroverkko.

Neuroverkot siis koostuvat yksittäisistä neuroneista. Kuten kuvasta 7 näkyy, neuronit on jaoteltuna kerroksittain. Neuronit ottavat sisääntulonsa edelliseltä kerrokselta ja antavat ulostulonsa seuraavalle kerrokselle. Kuvassa 7 oleva neuroverkko on täysin kytketty, monikerroksinen neuroverkko. Täysin kytketty tarkoittaa sitä, että jokainen kerroksen neuroni on kytketty jokaiseen seuraavan kerroksen neuroniin.

2.6.2 Neuroverkkojen arkkitehtuuri

Erilaisia neuroverkkoarkkitehtuureita on monenlaisia ja tilanteeseen sopivan arkkitehtuurin valinta riippuu monesta eri asiasta. Tässä kappaleessa keskitytään pelkästään kuvan 6 mukaiseen eteenpäin kytkettyyn neuroverkkoon, mutta on olemassa myös mm. takaisin-kytkettyjä neuroverkkoja ja konvoluutio neuroverkkoja.

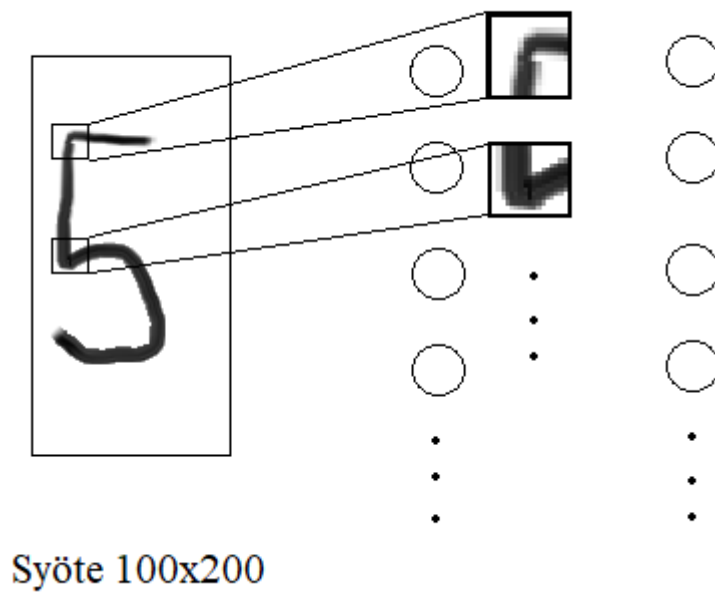
Otetaan esimerkiksi tapaus, jossa neuroverkkoa halutaan käyttää tunnistamaan, mikä numero kuvassa on kyseessä. Kyseessä on luokittelutehtävä, jossa kuvat tulee luokitella sen mukaan, mikä numero nolasta yhdeksään on kyseessä. Oletetaan, että käsin piirretyt numerot on irrotettu omaksi piirteeksi ja ne esitetään vaikka 10 x 20 pikselin harmaasävybittikarttana.

Kuvan kukin pikseli voidaan syöttää neuroverkon syötekerrokselle, eli tällöin tarvitaan 10×20 eli 200 neuronia syötekerrokselle. Kukin pikseli eli siten myös syötekerroksen vastaavat neuronit saavat arvon 0–255.

Ulostulona halutaan saada yksi kymmenestä luokasta, joka ilmaisee mikä numero on kyseessä. Valitaan tuloskerroksen kooksi kymmenen neuronia, yksi kutakin luokkaa kohden. Tuloskerroksen neuronit saavat desimaaliarvon väliltä 0–1, joka ilmaisee todennäköisyyden/varmuuden, jolla neuroverkon mielestä kuva esittää neuronia vastaavaa luokkaa. Esimerkiksi numeroa viisi vastaava neuronin voi saada arvokseen 0,96 ja muut tuloskerroksen neuronit alle 0,3 arvoja. Tällöin neuroverkko on vahvasti sitä mieltä, että kuvassa on numero viisi.

Esimerkin syöte- ja tuloskerrosten arkkitehtuuri on siis kohtalaisen yksinkertainen muodostaa. Piilokerrosten arkkitehtuurin valitseminen ei kuitenkaan ole niin yksinkertaista. Piilokerrosten lukumäärään tai kerroksella olevien neuronien määrään ei ole mitään yleispätevää sääntöä vaan parhaan arkkitehtuurin hakeminen on osittain kokeilemalla hakemista. Yleispätevänä sääntönä voi pitää, että mitä monimutkaisempi sisään- ja ulostulon suhde on, sitä useampi piilokerros ja sitä enemmän neuroneita piilokerrosta kohden. Yleensä syötekerroksella on enemmän neuroneita kuin tuloskerroksella, ja piilokerroksella olevien neuronien määrä pienenee lähestyttäessä tuloskerrosta. Piilokerrokselle on myös mahdollista laittaa enemmän neuroneita kuin syötekerrokselle. (Keim 2020; Sachdev 2020)

Piilokerrosten ja niillä olevien neuronien lukumäärän valitsemista esimerkissämme voidaan lähestyä valistuneen arvauksen kautta. Esimerkissämme on kyse kuvantunnistamisesta. Kuvantunnistamisessa, piilokerrosten tehtävänä voidaan ajatella olevan yksittäisten paikallisten piirteiden tunnistaminen. Tätä on havainnollistettu kuvassa 8. Seuraava kerros voi piirteistä saatujen tietojen perusteella päätellä korkeatasoisempia piirteitä ja lopulta sen mitä kuva esittää.



Kuva 8. Piilokerroksen neuronit voivat oppia tunnistamaan esimerkiksi paikallisia piirteitä.

Harmaasävykuvan numeron tunnistaminen on kohtalaisen yksinkertainen tunnistustehtävä, joten valitaan piilokerrosten määräksi yksi piilokerros. Mikäli ajatellaan piilokerroksen neuronien etsivän kuvasta piirteitä, neuronien määrä tulisi korreloida kuvasta etsittävien piirteiden määrään. Piirteiden määrää ei tosin pysty mitenkään järkevästi arvioimaan, joten kyseessä on jälleen vain arvio ja oikea neuronien määrä täytyy löytää kokeilemalla. Toisaalta neuronien määrällä ei ole niin suurta merkitystä eikä mitään yhtä oikeaa neuronien määrää ole. Piilokerroksen neuronien määräksi voi valita esimerkiksi 2/3 edellisen kerroksen koosta tai kaavalla

$$\sqrt{\text{syötekerroksen koko} \cdot \text{tuloskerroksen koko}}$$

Vaikka piilokerrosten ja niillä olevien neuronien lukumäärä onkin vapaasti valittavissa, ei määrää kannata tarpeettomasti kasvattaa. Tämä aiheuttaa turhaa kompleksisuutta ja vaikuttaa laskentatehoon. Lisäksi liian ison neuroverkon valitseminen aiheuttaa helposti ylisovittamista (*eng. overfitting*). Ylisovittamisella neuroverkko voi ”opetella ulkoa” opetusdatan ja toimia erinomaisesti opetusdatalla. Mutta kun ylisovitetulle neuroverkolle annetaan uutta oikeaa dataa, neuroverkko ei toimikaan hyvin. Ylisovittamisen ongelmat johtuvat siitä, että neuroverkko oppii yksityiskohtia, jotka ovat opetusdatan mukaisia, mutta

eivät ole todellisuudessa relevantteja todellisessa datassa. Neuroverkko siis oppii, että jokin sattumanvarainen ominaisuus olisi tärkeä määrittävä ominaisuus. (Russel & Norvig 2016, s. 705, 736)

2.6.3 Neuroverkkojen opettaminen

Neuroverkkojen opettaminen tarkoittaa käytännössä aktivaatiofunktiossa ja neuronien välillä olevien painotuksien w_{kj} sekä vakiotermien b_k säätämistä. Aluksi painotuksille annetaan sattumanvaraiset painoarvot. Neuroverkon opettaminen tapahtuu ns. vastavirta-algoritmin avulla (*eng. backpropagation*). Opettamisessa neuroverkolle annetaan sisään-tulona esimerkkitapauksia, joiden ulostulo on tiedossa. Neuroverkko antaa aluksi sattumanvaraisten painotusten vuoksi sattumanvaraisen ulostulon ja vertaa neuroverkon ulostuloa haluttuun ulostuloon. Vastavirta-algoritmin tehtävänä on minimoida neuroverkon ja todellisen ulostulon välinen erotus. Virheen laskemiseen käytetään virhefunktiota (*eng. loss function tai cost function*).

Vastavirta-algoritmin nimi tulee siitä, että algoritmi kulkee neuroverkkoa vastavirtaan tuloskerrokselta kohti syötekerrosta. Algoritmi katsoo neuroverkon ja todellisen ulostulon välistä virhettä ja kulkee neuroverkossa vastavirtaan päivittäen välien painotuksia paremmaksi. Algoritmin matemaattista toimintaperiaatetta ei käydä tarkemmin läpi, mutta algoritmin toiminta perustuu virhefunktion gradienttiin ja osittaisderivaattojen ketjutukseen. Ajatuksena on, että kukin neuroni vastaa pienestä osasta virhettä ja välien painotuksia muutetaan sen mukaan. (Russel & Norvig 2016, s. 733–735)

3 SISÄTILAPAIKANNUS JA UDELLLEENTUNNISTUS

Ulkotiloissa tapahtuva ihmisten paikannus alkaa olla tänä päivänä jo arkipäiväistä GPS:n (Global Positioning System) ansiosta. Sisätiloissa tapahtuva paikannus puolestaan on yhä suuri tutkimuksen aihe. GPS ei toimi luotettavasti seinien ja katon ympäröimässä sisätilassa, joten sisätilassa tapahtuvaan paikannukseen on olemassa muita vaihtoehtoisia tekniikoita. Kuitenkaan yhtä, joka tilanteeseen sopivaa menetelmää ei ole sisätilapaikannukseen löydetty. (Mautz 2012)

Sisätilapaikannuksen menetelmät voidaan jakaa kolmeen kategoriaan. (1) Liikkeeseen perustuvat metodit perustuvat esimerkiksi mukana kannettavaan sensoriin, joka mittaa kantajan liikettä ja seuraa sen avulla kantajan suhteellista paikkaa suhteessa aloituspisteeseen. Tämän tyyppinen metodi aiheuttaa herkästi suurta kumuloituvaa virhettä ja sitä käytetäänkin usein vain muiden metodien apuna (Jiang & Yin 2017). (2) Signaaliin perustuvat metodit ovat kaikista yleisimpiä menetelmiä toteuttaa sisätilapaikannus. Esimerkiksi WiFiin perustuvat menetelmät arvioivat mobiililaitteen kantajan paikkaa suhteessa WiFi-tukiasemiin signaalin vahvuuksien perusteella. Sisätilapaikannuksessa käytettäviä erilaisia signaaliin perustuvia menetelmiä on hyvin paljon, kuten Bluetooth, ZigBee ja RFID. WiFi on kuitenkin yksi yleisimmistä signaaliin perustuvista menetelmistä, koska WiFi-tukiasemat ovat jo hyvin yleisesti käytössä, eikä näin ollen ylimääräistä laitteistoa välttämättä tarvita. Kolmas kategoria sisätilapaikannukseen liittyen on (3) visuaalisuuteen eli kuviin perustuvat menetelmät. (Wei & Akinci 2019, s. 3)

3.1 Visuaalinen sisätilapaikannus

Myös visuaalisesta sisätilapaikannuksesta löytyy hyvin paljon tutkimusta. Yksi erittäin yleinen metodi on esimerkiksi mobiililaitteella otetun kuvan vertaaminen etukäteen tallennettuun kuvatietokantaan. Näin saadaan arvio, missä kyseinen mobiililaitte kuvanottohetkellä sijaitsee. Tällaista menetelmää voidaan käyttää esimerkiksi sisätilassa tapahtuvaan navigointiin ja opastukseen. Käyttäjä aktivoi mobiililaitteensa kameran ja saa paikkaan liitettyä informaatiota ja opastusta. Tämän tyyppinen sisätilapaikannus vaatii käyttäjän aktiivista toimintaa. (Kawaji ja muut 2010; Werner & Kessel 2011; Wei & Akinci 2019)

Sen sijaan, että paikannettava käyttäjä tuottaisi kuvamateriaalin ympäristöstä, ympäristössä oleva kamera voi tuottaa kuvamateriaalin paikannettavasta käyttäjästä. Eräs metodi, jolla saadaan ihmisten tarkka sijainti ja myös identifiointi luotettavasti tehtyä on tagiin perustuvat menetelmät. Tagi on jonkinlainen tunnistus, jota henkilö kantaa mukana. Valvontakameralla tehtävää tunnistusta varten tagi voi olla esimerkiksi infrapunavaloa emittoiva ledi tai sarja ledejä, jotka vilkuttavat ihmissilmälle näkymätöntä valoa tietyllä taajuudella. Kamera tunnistaa tagin lähettämän valon sekä taajuuden ja pystyy sen perusteella yksilöimään tagin. Tässäkin menetelmässä seurattavien henkilöiden on kannettava mukanaan ylimääräistä tunnistetta.

Viimeisenä menetelmänä on valvontakameran, konenäön ja tekoälyn avulla tehtävä henkilöiden tunnistus ja seuranta. Tässä menetelmässä ei tarvita muuta laitteistoa, kuin tilaa kuvaava kameralaitteisto sekä laskentatehoa.

Kameroiden välillä tapahtuvaa seurantaa on mahdollista toteuttaa kameroiden sijoittelun avulla. Sijoittamalla kamerat siten, että objektin poistuessa toisesta kamerakuvasta objekti tulee toisen kameran kuvaan, saadaan objektin radalle yhteys kameroiden välillä. Yleisempi on henkilön visuaalisten piirteiden avulla tehtävä tunnistus ja uudelleentunnistus. Tällöin kameroiden välille voi jäädä pidempiä katvealueita ja voidaan näin ollen katata laajempi alue.

3.2 Uudelleentunnistus

Uudelleentunnistuksella tarkoitetaan prosessia, jossa tunnistetaan, onko henkilö havaittu kameraverkostossa jo aikaisemmin (IGI Global 2021). Uudelleentunnistukseen kuuluu oleellisesti henkilön havainnointi, tunnistus ja seuranta videokuvassa. Nämä osa-alueet ovat itsessään jo hyvin laajalti tutkittua tutkimusaluetta (Nguyen ja muut 2016; Brunetti ja muut 2018). Viime aikoina myös uudelleentunnistuksen tutkimus on lisääntynyt monipuolisten sovellusmahdollisuuksien vuoksi (Zheng ja muut 2016; Pham ja muut 2017; Chen ja muut 2019).

Uudelleentunnistus voidaan jakaa kahteen pienempään ongelmaan. Ensimmäinen on henkilöiden piirteiden irrotus ja esittäminen merkitsevällä tavalla. Toinen on piirteiden avulla tehtävä luokittelu, jossa henkilöt voidaan tunnistaa ja erottaa toisistaan piirteistä saatavan

informaation avulla. Piirteiden irrotuksessa on mahdollista käyttää neuroverkkoja, mutta ne vaativat suuren datamassan opettamiseen ja ovat laskennallisesti raskaita. Chen ja muut (2019) suosittelevat mieluummin korkeatasoisten piirteiden irrottamista esimerkiksi värihistogrammin avulla ja luokittelemaan henkilöt piirteiden avulla yksilöityihin luokkiin.

3.3 Lentokenttäympäristöön tehty uudelleentunnistuksen toteutus

Camps ja muut ovat tehneet vuonna 2017 aiheeseen liittyvän tutkimuksen, jossa uudelleentunnistusongelmaa on viety laboratorioympäristöstä reaali maailmaan. Camps ja muut pyrkivät soveltamaan uudelleentunnistusongelmaa lentokenttäympäristössä. Tässä kappaleessa on kerrottu Campsin ja muiden tekemästä aikaisemmasta tutkimuksesta.

Camps ja muut suunnittelivat alusta alkaen järjestelmän, jolla on tarkoitus toteuttaa uudelleentunnistus useamman kameran välillä lentokenttäympäristössä. Suunnitteluun kuului luotettavan arkkitehtuurin suunnittelu, henkilön havainnointi ja seuranta sekä käyttäjystävällisen käyttöliittymän suunnittelu. Kamerainfrastruktuurina käytettiin lentokentällä jo käytössä olevaa kameraverkostoa. Campsin ja muiden tutkimuksessa kamerat oli aseteltu siten, että kamera A kuvasi aluetta, josta lentomatrustajat kävelivät ohi. Kamera A suoritti henkilöille identifioinnin, jonka jälkeen henkilöt tulivat jonkin ajan päästä joko kameraan B tai kameraan C. Uudelleentunnistus siis tehtiin joko kamerassa B tai kamerassa C.

3.3.1 Havainnointi, seuranta ja uudelleentunnistus

Ensimmäisenä huomion arvoisena asiana Campsin ja muiden tutkimuksesta voisi mainita henkilön havainnoinnin. Camps ja muut valitsivat henkilöiden havaitsemiseen Gaussin sekoitemallin (eng. *Mixtures of Gaussians, MoGs*) käyttämisen etualapikseleiden (henkilöiden) havaitsemiseen. Näin saadaan havaittua mielenkiintoiset alueet (eng. *Regions of Interests, RoI*), jotka voidaan rajata ympäröivillä laatikoilla. Jokainen kamera täytyi opettaa erikseen ja henkilön havainnointi perustui päätöspuu luokittelijaan (eng. *decision tree classifier*). Camps ja muut pitivät tärkeänä, että havaitseminen tapahtuu tehokkaasti noin 100 fps nopeudella.

Campsin ja muiden seuranta-algoritmi on kaksiosainen. Ensinnäkin havainnointi tehdään jokaisella kehyksellä edellä kuvatulla tavalla. Toiseksi seurannassa käytetään optiseen virtaan (*eng. optical flow*) perustuvaa ennustusmenetelmää, joka ennustaa liikevektorit edellisten kehysten perusteella ja liikevektorien perusteella ympäröivän laatikon todennäköisen nykytilan.

Uudelleentunnistuksessa Camps ja muut käyttivät piirrevektoreiden luomiseen väri- ja tekstuurihistogrammeja. Piirrevektoreiden avulla he pystyivät laskemaan kahden henkilön samankaltaisuuden. Heidän työssään on matemaattisesti tarkemmin kuvattu sitä, miten piirrevektoreista lasketaan kahden henkilön samankaltaisuus. Menetelmän valintaperusteena Campsilla ja muilla oli jälleen menetelmän laskennallinen tehokkuus.

3.3.2 Toteutuksessa vastaan tulleet haasteet

Campsin ja muiden tutkimuksen toisessa kappaleessa kerrotaan ongelmista, joita he kohtasivat suunnitellessaan ja toteuttaessaan reaali maailman uudelleentunnistusjärjestelmää. Ensinnäkin lentokenttäympäristön arkaluonteisen videomateriaalin vuoksi, järjestelmä täytyi pitää täysin eristyksissä internetistä. Tämä vaikeutti huomattavasti kehitystyötä, sillä sitä ei voinut tehdä etänä, vaan vaati lentokentällä käymistä fyysisesti.

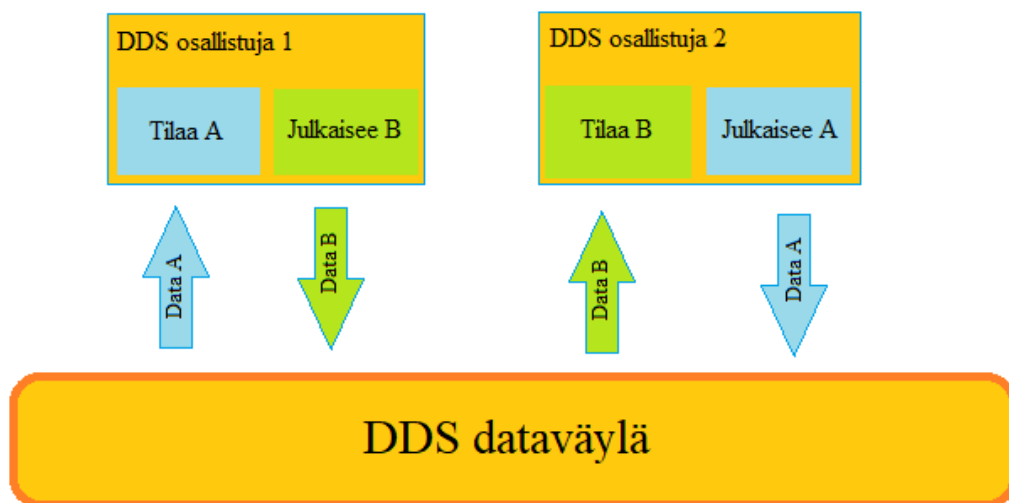
Toiseksi mainittiin joukko ongelmia, jotka liittyvät kameralaitteistoon. Koska kameralaitteistona käytettiin lentokentällä jo valmiiksi käytössä olevaa kameralaitteistoa, laitteisto sisälsi sekaisin analogisia ja digitaalisia kameroita. Analoginen videokuva täytyi muuntaa digitaalseksi erillisillä enkoodauslaitteilla. Myös kameroiden kuvanlaatu ei ollut kovinkaan hyvä. Lisäksi ongelmaksi voi muodostua valoisuusolosuhteiden vaihtelut etenkin ikkunoiden läheisyydessä sekä lattiasta tulevat kirkkaat heijastukset. Nämä mainitut ongelmat johtuvat suurimmaksi osaksi siitä, että Camps ja muut ovat joutuneet käyttämään lentokentän kiinteää kamerainfrastruktuuria.

Lisäksi haasteeksi voi muodostua etenkin ruuhkaisissa tiloissa seurattavien objektien jääminen toisten objektien taakse kameroiden näkymättömiin.

3.3.3 Arkkitehtuuri

Camps ja muut käyttivät arkkitehtuurina julkaisija-tilaaja -mallia (kuva 9.), jossa ohjelmiston eri moduulit tilaavat tarvitsemaansa dataa muilta moduuleilta ja julkaisevat omaa dataa muiden moduulien käyttöön. Käytössä heillä oli DDS-standardi (Data Distribution Service), jonka tehtävänä on datan hallinnointi moduulien välillä. Arkkitehtuurin valintaa perusteltiin mm. modulaarisuudella ja reaaliaikaisuusvaatimuksella. Kun järjestelmä on modulaarinen, voidaan osia vaihtaa helposti ja moduulit voivat toimia rinnakkain toisiinsa riippumatta.

Campsin ja muiden toteutuksessa moduuleina oli henkilöehdokkaiden havainnointi, henkilöehdokkaiden suodatus, piirteiden irrotus ja uudelleentunnistus. Lisäksi uudelleentunnistustulokset näytettiin graafisessa käyttöliittymässä.



Kuva 9. Julkaisija-tilaaja malli.

3.3.4 Datan kerääminen, käyttöliittymä ja uudelleentunnistustulokset

Camps ja muut tarvitsivat kohtalaisen paljon dataa, jonka avulla järjestelmää validoitiin ja arvioitiin. Lisäksi dataa käytettiin parametrien säätämiseen. Datan täytyi sisältää tarkasti oikeat henkilöiden havainnot. Käytännössä tämä tarkoitti sitä, että videokuvaan täytyi merkitä todelliset henkilöiden ympäröivät laatikot manuaalisesti. Tätä varten Camps ja muut olivat kehittäneet ohjelmiston, jolla videokuvaan merkitseminen onnistui kohtalaisella vaivalla.

Campsin ja muiden toteutuksessa käyttäjä merkitsi kamerassa A haluamansa henkilön ja käyttöliittymä näytti viisi piirteiltään lähimpänä olevaa henkilöä kamerasta B ja viisi piirteiltään lähimpänä olevaa henkilöä kamerasta C. Tämä vaikuttaa järkevältä toteutustavalta, sillä ohjelma harvoin arvioi oikean henkilön täsmälleen oikein. Useimmiten oikea henkilö löytyy jostain kymmenen todennäköisimmän henkilön joukosta. Ohjelma arvioi tunnistettavan henkilön täysin oikein vain noin 20 % tapauksista, kun taas tunnistettava henkilö oli viiden todennäköisimmän joukossa noin 60 % tapauksista ja kymmenen todennäköisimmän joukossa noin 85 % tapauksista.

3.3.5 Suositukset

Camps ja muut antavat työssään lopuksi joitain suosituksia reaali maailman uudelleentunnistusta varten. (1) He kehottavat panostamaan loppukäyttäjään, jotta järjestelmä olisi loppukäyttäjän kannalta toimiva ja helppokäyttöinen. (2) Järjestelmä on laskennallisesti vaativa, joten Camps ja muut kehottavat valitsemaan tehokkaan ohjelmistoarkkitehtuurin. (3) Hyvä objektin seuranta pienentää ylimääräisiä havaintoja ja parantaa suorituskkyä huomattavasti. (4) Viimeisenä suosituksena Camps ja muut mainitsevat loppukäyttäjän mukaan ottamisen front-endin suunnitteluun ja testaukseen mahdollisimman aikaisessa vaiheessa.

3.4 Torchreid-kirjasto uudelleentunnistukseen

Sopivaa ratkaisua etsittäessä löydettiin Torchreid-niminen tekoälysovelluskehikko. Torchreid on kehitetty syväoppimista ja uudelleentunnistuksen tutkimusta varten. Torchreid mahdollistaa erilaisten uudelleentunnistus mallien toteuttamisen, testaamisen ja vertailun. Torchreid:n avulla pystyy esimerkiksi testaamaan mallien toimivuutta eri dataseiteillä. Torchreid pitää sisällään valmiiksi 15 eri datasettiä sekä 20 eri mallia. (Zhou & Xiang 2019)

Torchreid on tarkoitettu eri mallien vertailua ja tutkimusta varten, mutta sen avulla on tehty myös valmiita uudelleentunnistusohjelmia. Esimerkiksi ”Samihorm” niminen Github-käyttäjä on tehnyt ohjelman, joka toteuttaa uudelleentunnistuksen videotiedostoille (Samihorm 2021). Samihormin ohjelmasta saadaan hyvää esimerkkiä, miten uudelleentunnistusta voidaan lähteä kokeilemaan Torchreid-kirjaston avulla. Haasteena on selvittää, miten uudelleentunnistus saadaan toimimaan reaaliaikaisen videokuvan kanssa.

4 TEKOÄLYN EETTINEN KEHITYS

Tekoälyä (ja uusia teknologioita yleisestikin) kehitettäessä on hyvä pohtia, mitä eettisiä haasteita uusi teknologia voi tuoda tullessaan. Tekoäly on yhä enenevässä määrin osa ihmisten jokapäiväistä elämää ja näin ollen sen seuraukset koskettavat monia elämän osa-alueita. Tekoäly voi tuoda paljon hyvää, kuten parantaa elämänlaatua ja luoda vaurautta, mutta se voi tuoda myös haasteita. Tekoäly voi toimia vinoutuneesti ja syrjiä esimerkiksi ihonvärin tai sukupuolen mukaan. Algoritmit ovat monesti täysin läpinäkymättömiä, jolloin kukaan ei todellisuudessa ymmärrä niiden todellista toimintaperiaatetta. Puhutaan niin sanotuista mustista laatikoista. Tekoälyn yhdessä muun teknologisen kehityksen kanssa pelätään monesti vievän työtä ihmisiltä, mutta se voi myös tehdä monesta työtehtävästä mielekkäämpää ja helpompaa, sekä luoda aivan uudenlaisia työtehtäviä. Toisaalta tekoäly voi laajentaa digitaalista kuilua ja kuilua pieni- ja suurituloisten välillä. Suurilla teknologiajäteillä on pääsy resursseihin, kuten suuriin datamassoihin, jotka mahdollistavat tekoälyn käytön. Tämä herättää kysymyksiä datan omistus- ja käyttöoikeuksista. Eräs korkealentoisempi eettinen uhkakuva tekoälyyn liittyen on ns. tekoäly singulariteetti, eli hetki, jolloin tekoäly ohittaa ihmisen älykkyydessään. Tämä herättää mielenkiintoisia kysymyksiä esimerkiksi tekoälyn autonomisuudesta sekä eksistentiaalisia kysymyksiä ihmislajin kannalta. Edellä mainitut haasteet ovat pieni joukko esimerkkejä. Eettiset haasteet liittyvät yleensä yksityisyyteen, läpinäkyvyyteen, luotettavuuteen, turvallisuuteen, taloudellisiin haasteisiin, oikeudenmukaisuuteen, vapauksiin sekä laajempiin sosiaalisiin haasteisiin. (Stahl 2021)

Tekoälyyn liittyviä eettisiä kysymyksiä on siis hyvin monenlaisia. Eettisiä kysymyksiä olisi hyvä pohtia aina, kun uutta teknologiaa lähdetään kehittämään tai ottamaan käyttöön. Varsinkin teknologian monimutkaistuessa eettiset haasteet eivät välttämättä tule kehitysvaiheessa ensimmäisenä mieleen. Eettisten kysymysten pohtiminen on kuitenkin hyvän yrityskulttuurin mukaista ja jossain määrin jopa edellytys. Kun yritys selvästi panostaa eettisyyteen, rahoituksen saaminen on helpompaa. Lisäksi asiakkaat ottavat tuotteita mieluummin käyttöön, kun he voivat luottaa yrityksen ja tuotteen toimivan eettisesti hyväksyttävällä tavalla. Mahdollisimman laaja läpinäkyvyys on yksi avain luottamuksen rakentamiseksi.

Yritykset voivat ottaa eettiset kysymykset osaksi jo olemassa olevaa normaalia riskienhallintasapluunaa. Ongelmana tässä vain on, että riskienhallinta ottaa yleensä huomioon vain kyseiseen organisaatioon liittyvät riskit. Eettisiä kysymyksiä pohtiessa tulisi ottaa huomioon myös muuhun yhteiskuntaan kohdistuvat riskit. (Stahl 2021, s. 62)

Yritykset ovat siis yksi taso, jolla tekoälyn eettisiin kysymyksiin voidaan panostaa. Muita tasoja ovat muun muassa lainsäätäjät, valtiot ja ylikansalliset järjestöt, kuten Euroopan unioni (Stahl 2021, s. 2). EU:n tekoälyä käsittelevä korkean tason asiantuntijaryhmä on laatinut laajan eettisen ohjeistuksen (2019). Ehkä suurin vastuu eettisistä kysymyksistä on kuitenkin yksittäisillä teknisillä asiantuntijoilla, jotka tekoälyä kehittävät ja edistävät. Heillä on mahdollisuus tehdä päätöksiä, jotka vaikuttavat suoraan siihen, miten eettisesti tekoälyä toteutetaan. Monesti saattaa tosin käydä niin, että tekoälyn kehittäjä ei huomaa pysähtyä miettimään tuotoksiensa eettisiä puolia. Ratkaisuna tähän voisi olla panostus jo koulutusvaiheessa teknologian varjopuoliin. Hyvänä esimerkkinä on tietoturva ja kyberturvallisuus laajemminkin. Kyberturvallisuuden pitäisi olla kaiken tietoteknisen opetuksen ja tekemisen pohjalla. Turvallisuutta on hankala lisätä jälkeenpäin, mutta kun tietoturva otetaan lähtökohtaisesti huomioon alusta alkaen, tietoturvan hoitaminen on paljon helpompaa. Jokaisen teknisen asiantuntijan olisi hyvä miettiä kaikessa tekemisessään jatkuvasti, mitä valitut ratkaisut tarkoittavat tietoturvan kannalta.

Yksilöt ja yritykset voivat siis tehdä paljon rakentaakseen eettisesti hyväksyttävää tekoälyä. Lisäksi lainsäädäntö ja regulaatio velvoittaa tietynasteisen etiikan noudattamista. Varsinkin Euroopassa kameravalvontaan ja tekoälyyn liittyvä regulaatio on erittäin laajaa (Haaranen & Allonen 2020).

Samaan aikaan kun Kiina on nousemassa tekoälyn edelläkävijäksi ilman eettisiä rajoituksia, Euroopan unioni painottaa enemmän yksilöiden oikeuksia ja hyviä etiikan toimintatapoja. Tämä ei kuitenkaan tarkoita sitä, että Euroopan tekoälykehityksen tarvitsisi pysähtyä eettisten pohdintojen vuoksi. Työ- ja elinkeinoministeriön julkaisemassa tekoälyohjelman loppuraportissa (2019, s. 35) painotetaan, että Euroopan on löydettävä tasapaino yksityisyyden suojan sekä kansalaisten oikeuksien ja tekoälyn hyödyntämisen välillä.

5 TEORIAOSUUDEN KÄYTÄNTÖÖN

YHTYMÄKOHDAT

TYÖN

Käytännön työn tarkoituksena on kehittää videokuvaa hyödyntävää uudelleentunnistusjärjestelmää. Järjestelmän avulla halutaan toteuttaa henkilöiden sisätilapaikannusta. Teoriaosuutta kirjoitettaessa saatiin paljon hyviä ajatuksia, jotka voidaan ottaa huomioon käytännön tekemisessä.

Konenäön teoriaan perehtyminen antoi hyvän pohjan sovelluskentän kokonaiskuvan hahmottamiseen. Lisäksi aiemmasta kirjallisuudesta saatiin hyödyllisiä tietoja esimerkiksi kameran valintaan ja sen vaikutuksesta kuvanmuodostukseen ja koko konenäköprosessiin. Erilaisten konenäköön liittyvien menetelmien osaaminen on hyödyksi, kun lähdetään kehittämään uutta konenäkösovellusta.

Teoriaosuuden toisessa osassa käytiin läpi sisätilapaikannuksen ja uudelleentunnistuksen teoriaa. Sisätilapaikannukseen liittyviä menetelmiä on useita, mutta tässä työssä tarkoituksena on keskittyä pelkästään visuaaliseen eli kameroiden avulla tehtävään sisätilapaikannukseen. Yleisimmät visuaalisuuteen perustuvat menetelmät ovat sovelluskohteita, joissa henkilöt kuvaavat esimerkiksi matkapuhelimellaan ympäristöä ja saavat sijaintinsa sitä kautta selville. Tämän työn sovelluskohteeseen sopii paremmin menetelmä, jossa kameraverkosto kuvaa tilaa ja paikallistaa kamerakuvassa näkyviä henkilöitä tekoälyn avulla.

Uudelleentunnistus osoittautui työtä kuvaavaksi ydintermiksi. Uudelleentunnistukseen perehtyminen antoi selkeän kuvan siitä, mitä käytännössä täytyy pyrkiä tekemään. Osaongelmia tulee olemaan uudelleentunnistettavien henkilöiden piirteiden irrotus ja esittäminen merkitsevällä tavalla sekä piirteiden avulla tehtävä henkilöiden erottaminen toisistaan. Vuonna 2019 Chen ja muut eivät suositelleet neuroverkkojen käyttämistä piirteiden luokitteluun. Syyksi mainittiin suurten datamassojen ja laskentakapasiteetin tarve. Tätä työtä varten löydettiin Torchreid-niminen ohjelmistokirjasto, joka perustuu juuri neuroverkkoihin. Torchreid pitää sisällään valmiiksi opetetut mallit, joten suuren datamassan tarve ei ole ongelma. Lisäksi viimeisten vuosien aikana on tapahtunut paljon kehitystä myös uudelleentunnistustutkimuksen alueella ja vuonna 2019 annettu suositus välttää neuroverkkoja ei välttämättä ole enää ajantasainen. Tänä päivänä suurin osa

uudelleentunnistukseen liittyvästä tutkimuksesta vaikuttaisi perustuvan neuroverkkojen käyttöön (Li ja muut 2021; Adaimi ja muut 2021; Tay & Yap 2021; Liu & Wu 2021).

Teoriaosuudessa käytiin läpi tutkimusta, jossa Camps ja muut (2017) ovat vieneet uudelleentunnistusongelmaa laboratorioympäristöstä reaalimaailmaan. Heidän tutkimukseensa uudelleentunnistusta pyrittiin soveltamaan lentokenttäympäristössä. Myös tämän työn tarkoituksena on soveltaa uudelleentunnistusta reaalimaailman ympäristössä. Campsin ja muiden tutkimus toimii hyvänä pohjana tälle työlle, sillä aiheet ovat hyvin lähellä toisiaan ja tutkimuksesta saadaankin valmiita askelmerkkejä tämän työn käytännön teke- mistä varten. Tutkimus on tehty vuonna 2017, joten menetelmät eivät välttämättä ole enää kaikilta osin ajan tasalla.

Camps ja muut käyttivät henkilöiden havaitsemiseen Gaussin sekoitemallia. Heidän menetelmänsä vaati jokaisen kameran erikseen opettamista henkilöiden havainnointiin. Henkilöiden seurantaan Camps ja muut käyttivät optista virtaa ennustamaan henkilöiden liikettä. Piirteiden irrotuksessa puolestaan käytettiin väri- ja tekstuurihistogrammeja.

Tässä työssä on tarkoituksena käyttää valmista ohjelmistoa, joka hoitaa kerralla henkilöiden havaitsemisen (Yolo-algoritmi) ja seurannan (DeepSORT-algoritmi). Ohjelmisto käyttää havaitsemiseen ja seurantaan neuroverkkoihin perustuvia menetelmiä ja on näin ollen laskennallisesti raskaampi, kuin Campsin ja muiden menetelmät. Niinpä kannattaa- kin pitää mielessä Campsin ja muiden vaihtoehtoinen tapa, mikäli sovelluksen kehityk- sessä suorituskyky osoittautuu ongelmaksi.

Yolo-algoritmi irrottaa suoraan myös havaittujen henkilöiden piirteet. Tässä työssä on mahdollista kokeilla, voidaanko Yolo-algoritmilta saatuja piirteitä käyttää myös uudelleentunnistukseen. Toisaalta Torchreid-ohjelmistokirjasto mahdollistaa myös piirteiden irrotuksen.

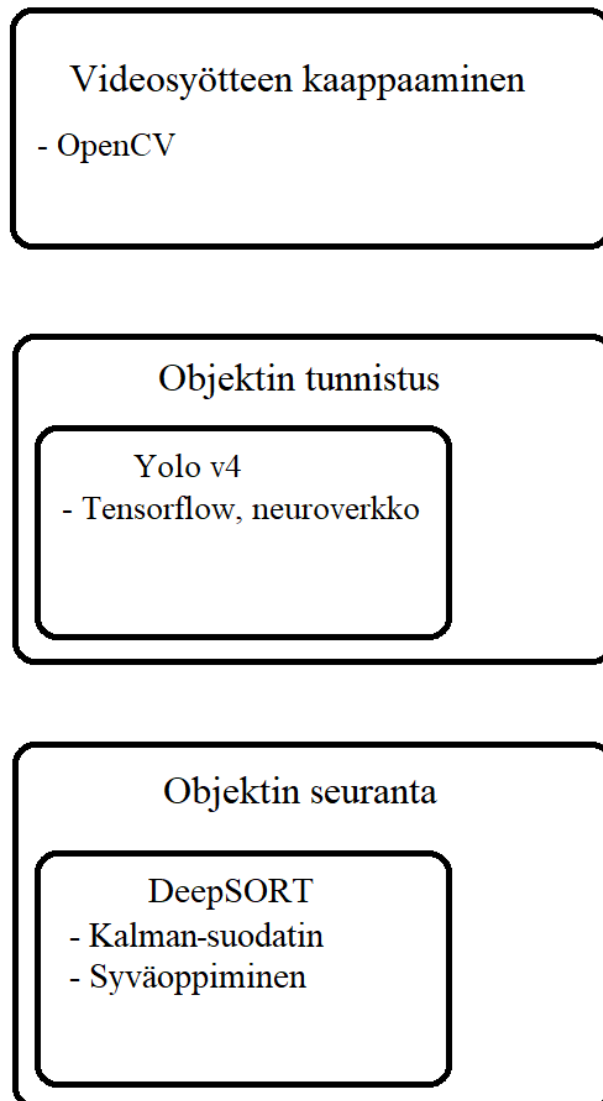
Camps ja muut kertoivat myös haasteista, joita he tutkimuksessaan kohtasivat. Arkaluon- toisen videomateriaalin vuoksi kehitystyötä ei voinut tehdä internetin välityksellä etänä vaan vaati käyntiä paikan päällä. Tämä saattaa muodostua osittain haasteeksi myös tässä työssä.

Toiseksi Campsilla ja muilla oli joukko haasteita, jotka liittyivät lentokenttäympäristön kiinteään kameraverkostoon. Toivon mukaan tässä työssä voidaan minimoida näitä ongelmia valitsemalla laadukasta kuvaa tuottavat kamerat, jotka ovat säädettävissä pitch- ja yaw-suunnassa, sekä sijoittamalla kamerat optimaalisella tavalla.

Campsin ja muiden toteuttama tapa tehdä ohjelmistosta modulaarinen vaikuttaa järkevältä. Campsin ja muiden toteutuksessa eri moduuleita oli henkilöhdokkaiden havainnointi, henkilöhdokkaiden suodatus, piirteiden irrotus ja uudelleentunnistus. Tässä työssä on tarkoituksena kokeilla hyödyntää Yolo -algoritmia, joka toimii mahdollisesti yhtenä suurena moduulina, joka hoitaa suoraan henkilöiden tunnistuksen ja piirteiden irrotuksen.

6 OBJEKTIN TUNNISTUS- JA SEURANTAOHJELMA

Tässä työssä kehitettävän ohjelmiston pohjana toimii avoimeen lähdekoodiin perustuva YOLOv4-deepsort-ohjelmisto (The AI Guy 2021). Tässä kappaleessa esitellään alkuperäisen lähdekoodipohjan toimintaperiaatetta tärkeimpien komponenttien osalta. Kuvassa 10 on kuvattu YOLOv4-deepsort-ohjelmiston toimintaperiaatteen kannalta oleelliset komponentit.



Kuva 10. YOLOv4-deepsort-ohjelman tärkeimmät komponentit/teknologiat.

Yolov4-deepsort-ohjelmisto on Python-ohjelmointikielellä kirjoitettu ohjelmisto, joka toteuttaa videokuvassa olevien objektien tunnistuksen ja seurannan. Kuten kuvasta 8 näkee, videosyötteen kaappaamisesta vastaa OpenCV-niminen (Open Source Computer Vision Library) Python-kirjasto. OpenCV on myös avoimeen lähdekoodiin perustuva, lähinnä reaaliaikaisia konenäkösovelluksia varten tehty ohjelmakirjasto (OpenCV team 2021). Videosyötteen voi kaapata OpenCV-kirjaston avulla ohjelmaan esimerkiksi webbikamerasta tai verkon välityksellä antamalla videolähteen osoitteen. OpenCV:n rooli ohjelmassa on myös mm. käsitellä videokuvan yksittäiset kehykset, tehdä esikäsittely toimintoja sekä piirtää objektien ympäröivät kehikot.

Objektin tunnistuksesta vastaa Yolo v4 (You Only Look Once) niminen algoritmi. Kyseessä on nimensä mukaan neljäs versio Yolo-algoritmista. Yolo- objektintunnistusalgoritmin vahvuutena on sen nopeus, joka mahdollistaa reaaliaikaisen objektintunnistuksen videokuvasta jopa 60 (fps, frames per second) kehyksen sekuntinopeudella (Bochkovskiy ja muut 2020). Yolo-algoritmin toimintaperiaate on melko monimutkainen eikä sitä käydyä tässä läpi kuin pääpiirteittäin.

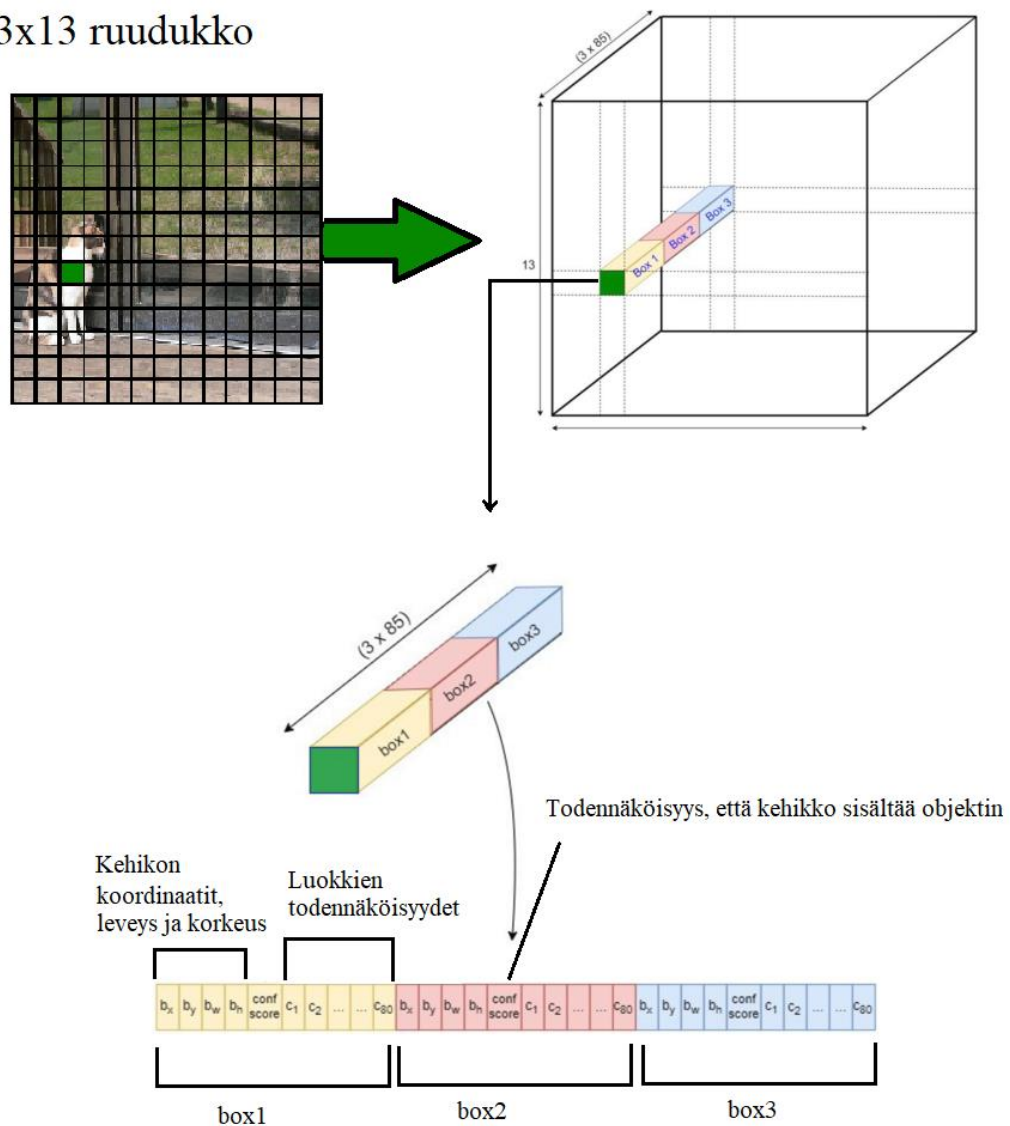
Ohjelmistossa on käytetty hyödyksi Googlen Tensorflow-kirjastoa, jonka avulla toteutetaan konvoluutio neuroverkko. Neuroverkko on oleellinen osa Yolo-algoritmia. Yolo-algoritmista neuroverkolla on yhteensä 106 kerrosta. Objektien havainnointi tapahtuu kerroksilla 82, 94 ja 106. Näillä kerroksilla alkuperäinen kuva on skaalattuna kolmeen eri kokoon, joten näin kuvasta saadaan tunnistettua eri kokoisia objekteja. (Sichkar 2020)

Yolo-algoritmi jakaa kuvan $S \times S$ -kokoiseksi ruudukoksi. Kunkin ruudun tehtävä on tunnistaa, kuuluuko jonkin objektin keskikohta ruudun sisään. Koska tunnistus tapahtuu kolmella eri kerroksella, kolmelle eri kokoon skaalatulle kuvalle, $S \times S$ ruudukon koko on näillä kerroksilla vastaavasti 13×13 , 26×26 ja 52×52 .

Yolo-algoritmi tunnistaa kuvassa olevien objektien ympäröivät kehikot ja luokittelee objektit niiden piirteiden avulla. Ohjelmaa varten on olemassa etukäteen opetetut painotukset, jotka tulee muuntaa Tensorflow-neuroverkolle sopivaan muotoon. Painotukset on opetettu niin kutsutulla COCO-datasetillä (Common Objects in Context), jossa on 80 yleistä objektiluokkaa, kuten auto, jalankulkija, koira jne. (Lin ja muut 2015).

Kullekin $S \times S$ ruudukon ruudulle lasketaan $b * (5 + c)$ attribuuttia, jossa b on ennalta määrätty erikokoisten ympäröivien kehikoiden määrä (esimerkiksi 3), ja c on objektiluokkien määrä (COCO-datasetillä 80). Loput attribuutit ovat ympäröivän kehikon x -, ja y -koordinaatit, ympäröivän kehikon leveys ja pituus sekä todennäköisyys, että objektin keskikohta on kyseisessä ruudussa. $S \times S$ ruudukkoa ja attribuutteja on havainnollistettu kuvassa 11.

13x13 ruudukko



Kuva 11. Yolo-algoritmin objektin tunnistuksessa käytetyt attribuutit havainnollistettuna (Kuva julkaistu Creative Commons -linsessillä. COCO Consortium 2017).

Yolo-algoritmin tehtävänä on siis paikallistaa objektien ympäröivät kehikot sekä arvioida mikä objekti on kyseessä. Ympäröivät kehikot syötetään ohjelmassa eteenpäin objektin seurantaan.

Objektin seurannasta ohjelmassa vastaa DeepSORT (*eng. Simple Online Realtime Tracking with a Deep Association Metric*) -niminen algoritmi. Objektin seuranta varmistaa ohjelman luotettavamman tunnistuksen, ettei tunnistuksesta tule katkonainen. Seuranta-algoritmin toimintaperiaatetta ei käydä tarkemmin läpi, sillä se ei ole sovelluksen kehityksen kannalta niin olennaisessa osassa kuin esimerkiksi tunnistusalgoritmi. Seuranta-algoritmi saa tunnistusalgoritmilta havainnot objekteista ja päivittää ennustetta seuraavista havainnoista. DeepSORT-algoritmin tärkein objekti on niin kutsutut ”radat”. Rata tarkoittaa kunkin havaitun objektin kulkemaa rataa ja eräitä siihen liittyviä attribuutteja. Radan ”elinikää” voidaan esimerkiksi muuttaa sen mukaan, miten kauan rataa halutaan säilyttää muistissa ilman uutta varmaa havaintoa objektista. Lisäksi Yolo-algoritmilta saadut piirteet säilytetään radan muistissa.

7 SOVELLUKSEN KEHITTÄMINEN

Teoriaan, sovelluskohteeseen ja valmiina oleviin ohjelmistoihin perehtymisen jälkeen aloitettiin sovelluskohteen käytännön kehittäminen. Sovellus vaatii kohtalaisen paljon laskentatehoa, joten sovellusta varten on hankittu erillinen Linux-palvelin. Sovelluskohteen luonteen vuoksi palvelimella on käytössä kaksi Nvidia-merkkistä näytönohjainta, jotka kykenevät vaatimaan rinnakkaiseen laskentaan. Näytönohjaimia käytetään pääasiassa graafiseen laskentaan sekä myös tekoälysovelluksiin juuri rinnakkaisen laskentakykynsä vuoksi (Feng ja muut 2019 s. 315).

7.1 Kehitysympäristön käyttöönotto ja testaus

Ensimmäisenä tehtävänä oli saada kehitysympäristö pystyyn ja toimivaksi. Tavanomaisella kannettavalla tietokoneella kehitystyö on haastavaa, sillä ohjelma ei toimi kunnolla puutteellisen laskentatehon vuoksi. Niinpä Linux-palvelin oli saatava käyttöön heti alusta alkaen.

Linux-palvelin on hankittu oululaisesta pilvipalveluja tarjoavasta yrityksestä. Palvelimelle kirjaudutaan etänä, jonka jälkeen voidaan testata olemassa olevan ohjelmiston toimintaa. Ohjelman toimintaa on kuvattu tarkemmin kappaleessa neljä. Sen pääasiallinen tehtävä on tunnistaa videosyötteestä objekteja. Videosyöte ohjelmaan otetaan tässä alkuvaiheessa Youtube-streamista. Ohjelman tehtävä on näyttää videokuva, johon on merkitty havaitut objektit. Koska etäyhteys Linux-palvelimelle tapahtuu tekstimuotoisesti SSH-yhteydellä, täytyy videokuva saada jollakin muulla tavalla kehitystyötä varten. Tämä on toteutettu yksinkertaisella palvelinohjelmistolla, jolloin ohjelman tuottamaa videokuva pystyy katselemaan nettiselaimella.

Kehitysympäristöön kuuluu siis Linux-palvelin, jolla objektintunnistusohjelmistoa ajetaan. Samalla Linux-palvelimella ajetaan myös yksinkertaista palvelinohjelmistoa, joka antaa mahdollisuuden tarkastella tuotettua kuvaa nettiselaimen välityksellä. Selaimen kautta välitetty videokuva toimii melko hitaasti, mutta kuitenkin riittävän hyvin, että kehitystyö voidaan aloittaa.

Ensimmäisenä arvioidaan jo olemassa olevan ohjelmiston toimivuutta. Ohjelma havaitsee henkilöitä videosyötteestä kohtalaisen hyvin, mutta se tekee myös vääriä havaintoja, jotka eivät oikeasti ole henkilöitä. Uudelleentunnistuksen kannalta on parempi, että ohjelma havaitsee vääriä positiivisia, kuin jättää henkilöitä havaitsematta.

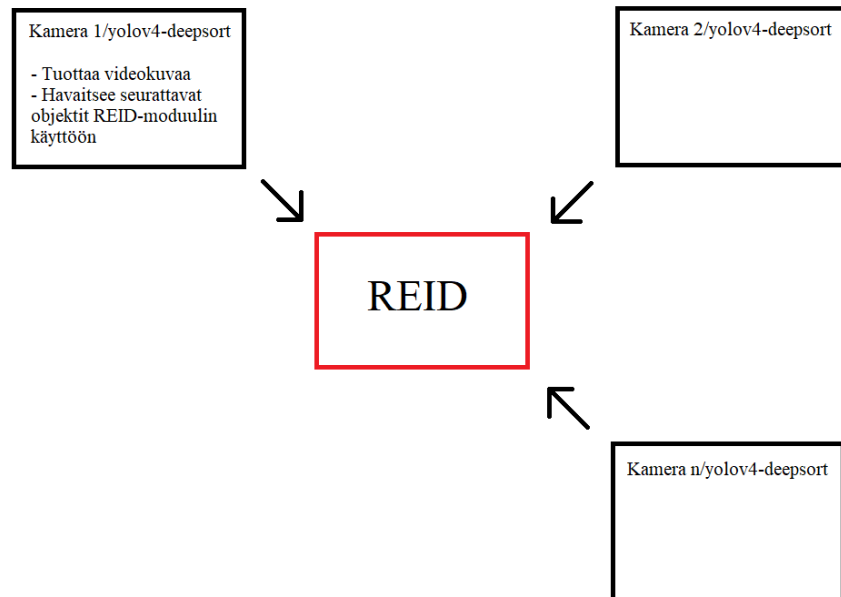
Ohjelma merkitsee kunkin havaitun henkilön ympäröivän laatikon yhteyteen: ”Henkilö1”, ”Henkilö2”, jne. Mikäli henkilön havaitseminen katkeaa pidemmäksi aikaa, ja henkilö tunnistetaan hetken päästä uudelleen, henkilön numero muuttuu. Ohjelma kykenee siis luomaan havaituille henkilöille yksilöivän numeron, mutta mikäli henkilön havaitseminen ja seuranta katkeaa hieman pidemmäksi aikaa, sama henkilö yksilöidään eri henkilöksi.

Kun ohjelmistoa on testattu Youtube-streamin avulla, seuraava tehtävä on asentaa kamera tuottamaan videokuvaa ohjelmiston käyttöön. Kameraksi valikoitui Eufy Security IP-kamera. Kamerasta täytyy asettaa päälle RTSP-protokollalla (Real-Time Streaming Protocol) toimiva videokuvan jakaminen. Tämä mahdollistaa videokuvan kaappaamisen OpenCV-kirjaston avulla Python-ohjelmiston käyttöön samassa verkossa olevalle tietokoneelle. Koska videokuva täytyy saada pilvipalvelussa olevalle Linux-palvelimelle, täytyy kameralin verkon palomuurin tehdä porttiohjaus ja sallia sisään tuleva liikenne Linux-palvelimen osoitteesta. Näin videokuva saadaan internetin välityksellä Linux-palvelimelle RTSP-protokollan avulla.

7.2 Uudelleentunnistuksen kehittäminen

Kun testiympäristö on saatu valmiiksi ja kamerakuva siirrettyä Linux-palvelimella ajettavan ohjelmiston käyttöön, voidaan aloittaa varsinaisen uudelleentunnistusohjelman kehittäminen. Ohjelmiston pohjana toimii Yolov4-deepsort-ohjelmisto, joka tunnistaa videokuvasta seurattavat henkilöt. Uudelleentunnistusongelmaa lähestytään suunnittelemalla mahdollista arkkitehtuuria ohjelmistolle. Suunnittelun apuna käytetään työn teoriaosuudessa vastaan tulleita aiempia tutkimuksia ja niistä kertyneitä oppeja. Etenkin Campsin ja muiden tutkimuksessa ollut tapa tehdä ohjelmistosta modulaarinen vaikuttaa lupaavalta käytännöltä.

Kuvassa 12 on kuvattu ohjelman arkkitehtuurin modulaarisuutta. Ohjelmiston keskiössä on uudelleentunnistuksesta vastaava uudelleentunnistusmoduuli (REID). Tähän uudelleentunnistusmoduuliin voi liittyä n -kappaletta videokuvaa tuottavia kamera-/Yolov4-deepsort-moduuleita. Reid-moduulin tehtävänä on tehdä uudelleentunnistus myös eri kameroiden välillä.



Kuva 12. Ohjelmiston modulaarisuus.

Kehityksen alkuvaiheessa löytyi ”Samihorm” Github-käyttäjän tekemä uudelleentunnistushjelmisto. Samihormin ohjelmiston lähdekoodi on vapaasti saatavilla ja sitä tutkimalla voi saada konkreettisen kuvan siitä, miten uudelleentunnistuksen voisi toteuttaa. Samihormin ohjelmistossa on käytetty taustalla Torchreid-ohjelmistokehikkoa. Torchreid toteuttaa ohjelmassa tekoälyn, joka vastaa henkilöiden piirteiden luomisesta ja vertailusta. Samihormin ohjelmisto on tarkoitettu toteuttamaan uudelleentunnistus videotiedostoille, joten ohjelmistoa ei voi hyödyntää suoraan. Ohjelmistosta voidaan kuitenkin ottaa mallia uudelleentunnistuksen toteuttamiseen myös reaaliaikaisessa videostreamissa.

Torchreid-ohjelmistokehikkoon perehtyminen ja sen käyttöönotto vaativat kohtalaisen paljon työtä. Torchreid tarjoaa kuitenkin valmiita työkaluja uudelleentunnistusta varten, joten sen käyttäminen on selkeä valinta.

Aluksi Torchreid:n toimivuutta testataan liittämällä sen toiminnallisuus suoraan Yolov4-deepsort-ohjelmaan, eli siis ilman modulaarista rakennetta. Alkuvaiheessa tavoitteena on saada uudelleentunnistus toimimaan edes yhdessä kamerassa eikä useamman kameran järjestelmää vielä tässä vaiheessa pohdita. Torchreid:n tehtävä on jokaisella kehyksellä irrottaa havaittujen henkilöiden piirteet sekä vertailla piirteiden samankaltaisuutta. Ohjelmisto saatiin toimimaan, mutta sen suorituskyky on niin heikko, ettei ohjelmaa voi pitää tosiasiallisesti toimivana. Ohjelmiston ruudunpäivitysnopeus putosi reilusti alle yhteen kehykseen sekunnissa.

Seuraavaksi kokeillaan saadaanko suorituskykyä parannettua tekemällä ohjelmistosta modulaarinen. Yolov4-deepsort-ohjelmisto voisi toimia omana prosessinaan (lähes) samalla tavalla kuin ilman uudelleentunnistustakin ja uudelleentunnistusohjelmisto toimisi omana prosessinaan. Näin ollen uudelleentunnistus ei suuremmin vaikuttaisi Yolov4-deepsort-ohjelmiston suorituskykyyn. Täytyy vain kehitellä jokin järkevä tapa, millä havaittujen henkilöiden piirteet saadaan järkevästi siirrettyä kahden eri prosessin välillä.

Aikaisemmin tässä työssä pohdittiin, voidaanko Yolov4-deepsort-ohjelman tuottamia objektien piirteitä käyttää suoraan hyödyksi uudelleentunnistuksessa. Yolov4-deepsortin piirteet ovat 128:n lukuarvon kokoisia vektoreita. Samiormin tekemä ohjelmisto yhdessä Torchreid:n kanssa mahdollistaa myös oman tapansa piirteiden irrotukseen. Tässä tapauksessa piirteet ovat 1x2048 kokoinen tensori, eli siis selvästi tarkempi piirrekuvaus kuin Yolov4-deepsortin piirrekuvaus. Lisäksi on mahdollista, ettei Yolov4-deepsortin tuottamat piirteet ole yhteensopivia Torchreid:n vertailun kanssa. Tämän vuoksi päätettiin käyttää Torchreid:n piirteitä Yolov4-deepsort-piirteiden sijaan.

Piirteet täytyy siis saada jollain tavalla uudelleentunnistusprosessin käyttöön Yolov4-deepsort-ohjelmiston havaitsemista henkilöistä. Tämä päätettiin toteuttaa siten, että piirteet irrotetaan Torchreid:n avulla Yolov4-deepsort-ohjelmistossa. Suorituskyvyn parantamiseksi piirteiden irrotusta ei tehdä jokaisella kehyksellä, vaan esimerkiksi joka kymmenennellä kehyksellä. Ohjelmassa on helposti säädettävissä, kuinka usein piirteiden irrotus tehdään. Tällä parametrilla voidaan säätää ohjelman suorituskyvylle ja tarkkuudelle sopiva kompromissi. Mitä useammalla kehyksellä piirteiden irrotus tehdään, sitä paremmin uudelleentunnistuksen pitäisi toimia, mutta toisaalta laskennallinen taakka kasvaa ja näin ollen videokuvan ja ohjelmiston ruudunpäivitysnopeus putoaa.

Kun piirteet irrotetaan YOLOv4-deepsort-ohjelmassa Torchreid:n avulla, ne tallennetaan tiedostopohjaiseen SQLite3-tietokantaan. Tällä tavalla uudelleentunnistusprosessi voi hakea tietokannasta piirteet omaan käyttöön. Tietokantaan kirjoittaminen ja sieltä lukeminen vaativat myös jonkin verran laskentatehoa. Kokeilemisen kautta todettiin, ettei suorituskyky heikkene merkittävästi, jos piirteiden irrotus ja tietokantaan kirjoittaminen tehdään joka kymmenennellä kehyksellä. Henkilön piirteen kanssa tietokantaan tallennetaan myös YOLOv4-deepsort-ohjelman tuottama henkilön tunniste (ID) sekä mahdollisesti muita haluttuja tietoja, kuten aikaleima, sekä monen kameran järjestelmässä kameran tunniste.

Kun piirteiden irrottaminen sekä siirtäminen uudelleentunnistusohjelman käyttöön on todettu toimivaksi, seuraava tehtävä on suunnitella ja toteuttaa uudelleentunnistusmoduuli. Uudelleentunnistuksen perustoimintaperiaate on seuraava:

1. Ota tietokannasta uusimman havaitun henkilön piirteet ja ID.
2. Mikäli henkilön ID on uudelleentunnistusohjelmalle tuttu, lisätään uusi piirretensori kyseiselle ID:lle muistiin.
3. Mikäli henkilön ID on uudelleentunnistusohjelmalle tuntematon, verrataan henkilön piirteitä tunnettujen henkilöiden piirteisiin. Mikäli uudet piirteet ovat riittävän lähellä jotakin toista piirrejoukkoa, ohjelma ilmoittaa epäilevänsä kahden ID:n olevan sama henkilö.

Henkilöiden piirteiden vertaaminen tapahtuu Torchreid:n avulla. Torchreid tarjoaa valmiin funktion, joka laskee kahden piirretensorijoukon etäisyysmatriisin. Torchreid:ssä on mahdollista valita laskentatavaksi Euklidisen etäisyyden neliö tai kosini etäisyys. Käytettäväksi etäisyydeksi valittiin Euklidisen etäisyyden neliö, joka on Torchreid:ssä käytössä oletuksena. Etäisyysmatriisista otettiin keskiarvo, jolloin etäisyydelle saadaan yksittäinen lukuarvo.

8 TESTIT JA ANALYSOINTI

Kun uudelleentunnistusmoduulin ensimmäinen prototyyppi on saatu valmiiksi, on aika testata sen toimivuutta. Testiympäristönä on toimiston käytävä, jonka toisessa päässä on IP-kamera. Kannettavan tietokoneen avulla voidaan ottaa etäyhteys Linux-palvelimelle, jolla YOLOv4-deepsort-ohjelmistoa ja uudelleentunnistusmoduulia ajetaan. YOLOv4-deepsort-ohjelmisto kaappaa IP-kameran videokuvan internetin välityksellä, havaitsee videokuvasta henkilöitä ja tallentaa tietokantaan havaittujen henkilöiden tunnistukset sekä piirteet. Uudelleentunnistusmoduuli puolestaan lukee tietokannasta havaitut henkilöt, tekee niille uudelleentunnistuksen ja tulostaa uudelleentunnistustulokset kannettavan tietokoneen etäkonsoliin.

8.1 Ensimmäinen testaus

Ensimmäisellä testikerralla kokeillaan vain kävellä käytävällä lyhyt matka, mennä hetkeksi kameran näkymättömiin ja kävellä hetken päästä sama matka uudestaan. YOLOv4-deepsort-ohjelmisto antaa tällä tavalla hetken näkymättömissä olleelle henkilölle kaksi eri tunnistetta ja näin ollen voidaan testata, minkä etäisyyden arvon uudelleentunnistus antaa samalle henkilölle.

Seuraavaksi sama testi toistetaan, mutta tällä kertaa kameran kuvassa käy kävelemässä kaksi erinäköistä henkilöä. Näin voidaan verrata minkä etäisyyden arvon uudelleentunnistus antaa kahdelle erinäköiselle henkilölle ja minkä etäisyyden arvon kahdelle samankäköiselle henkilölle. Näin voitiin ensimmäistä kertaa arvioida uudelleentunnistuksen toimivuutta.

Edellä mainittu testi ei tuottanut hyviä tuloksia, vaan kahden piirrejoukon etäisyysarvot olivat melko kaukana toisistaan riippumatta siitä, oliko kyseessä oikeasti sama vai eri henkilö.

Ensimmäisellä kerralla tehty kontrolloimaton kävely ei tuottanut halutun kaltaisia tuloksia, joten seuraavaksi kokeillaan suorittaa samankaltainen testi, mutta kontrolloidumalla tavalla. Tällä kertaa henkilö ei kävele sattumanvaraisesti käytävällä, vaan ilmestyy nopeasti kamerakuvaan, on hetken aikaa kohtalaisen paikallaan ja poistuu nopeasti

kamerakuvasta. Näin henkilön piirrejoukko pysyy koko ajan mahdollisimman samankaltaisena.

Tällä tavalla kontrolloidulla testillä tulokset olivat jo paljon parempia. Kahden piirrejoukon välinen etäisyys oli huomattavasti pienempi, mikäli kyseessä oli oikeasti sama henkilö. Uudelleentunnistus vaikutti siis toimivan. Taulukossa 1 on kontrolloidun testin tulokset. Luvut kuvaavat kahden ID:n välistä etäisyyttä toisistaan. Luvut ovat laaduttomia ja pyöristetty yhden desimaalin tarkkuudella. Toisaalta luvut olisi mahdollista pyöristää myös kokonaisluvuiksi tai jopa kymmenien tarkkuudella. Luvut eivät itsessään kerro juuri mitään, vaan lukuarvoja täytyy verrata toisiin etäisyysarvoihin. Etäisyysarvot myös vaihtelevat eri testien välillä eri parametreista riippuen, joten luvut eri testien välillä eivät ole suoraan verrattavissa toisiinsa. Oleellisia kysymyksiä on, ”miten etäisyyden arvo riippuu siitä, onko kyseessä samannäköiset vai erinäköiset yksilöt” sekä ”onko löydettävissä selkeä jaottelu samannäköisten ja erinäköisten yksilöiden etäisyyksien välillä”. Taulukossa vihreällä merkityt ruudut tarkoittavat etäisyyttä samannäköisten henkilöiden välillä ja punaisella merkityt ruudut etäisyyttä erinäköisten henkilöiden välillä. Kuten tuloksista nähdään, samannäköisten henkilöiden välinen etäisyys on alle 100, kun taas erinäköisten henkilöiden välinen etäisyys on yli 250. Näin ollen voidaan tehdä jaottelu, että alle 100 etäisyyden arvolla henkilöt ovat samannäköisiä ja yli 250 etäisyydellä henkilöt ovat erinäköisiä. Mitä suurempi jaottelun marginaali on, sitä parempi. Uudelleentunnistusohjelmaan on mahdollista määrittää tietty raja-arvo, jonka alle menevillä etäisyyden arvoilla yksilöt luokitellaan samannäköisiksi. Esimerkiksi taulukon 1 tulosten perusteella raja-arvoksi voitaisiin valita 175, joka on marginaalin puolivälissä.

Taulukko 1. Kahden ID:n väliset etäisyydet kontrolloidussa testissä.

	Punainen paita ID 1	Punainen paita ID 2	Punainen paita ID 3	Musta takki ID 4	Musta takki ID 5	Punainen paita ID 6
Punainen paita ID 1						
Punainen paita ID 2	40.2					
Punainen paita ID 3	33.1	36.9				
Musta takki ID 4	295.0	295.6	309.9			
Musta takki ID 5	355.4	344.2	370.8	98.6		
Punainen paita ID 6	29.6	30.1	62.4	283.5	380.5	

8.2 Sovelluksen parantaminen ja uusi testaus

Sovellus siis vaikuttaa toimivan hyvin kontrolloiduissa olosuhteissa, mutta mikäli sovellusta yritetään käyttää hiemankin todellisuutta vastaavissa olosuhteissa, uudelleentunnistus ei vaikuta toimivan. Seuraavaksi pohdittiin syitä, miksi uudelleentunnistus ei toimi reaalisissa olosuhteissa sekä parannuskeinoja, joilla uudelleentunnistus saataisiin toimimaan paremmin.

Ensimmäisenä ongelmana tulee mieleen uudelleentunnistuksen toteuttaminen pelkästään tuntemattomille henkilöille. Kun henkilö havaitaan ensimmäisen kerran, henkilölle tehdään uudelleentunnistus heti ensimmäisen havaitun piirretensorin avulla. Ohjelmaa voitaisiin parantaa keräämällä joukko piirteitä ja tekemällä uudelleentunnistus vasta, kun

piirteitä on haluttu määrä. Tunnistuksen voi loogisesti ajatella olevan sitä luotettavampi, mitä suurempi vertailtavien piirteiden joukko on.

Toiseksi Yolov4-deepsort-ohjelma irrotti henkilöiden piirteet joka kymmenennellä kehyksellä. Tätä parametria olisi mahdollista säätää siten, että piirteet irrotetaan, vaikka joka viidennellä kehyksellä. Ohjelmasta tulee tällöin laskennallisesti raskaampi, joten ohjelman riittävä suorituskkyky täytyisi varmistaa. Kun henkilöiden piirteet irrotetaan useammalla kehyksellä, henkilöiden piirrejoukot kasvavat vastaavasti nopeammin. Toisaalta henkilön asento ja näin ollen piirteet eivät ehdi muuttua niin paljoa piirteiden irrotuksen välillä.

Niinpä seuraavaksi lähdettiin etsimään sopivaa kompromissia piirteiden irrotusnopeuden ja uudelleentunnistuksessa käytettävän piirrejoukon koon välillä. Piirteiden irrotusnopeutta kasvatetaan siten, että piirteet irrotetaan joka viidennellä kehyksellä. Lisäksi uudelleentunnistuksen vertailu tehdään vasta, kun henkilön piirrejoukon koko on kymmenen. Tällöin kymmenen piirteen irrottamiseen tarvitaan vähintään 50 kehystä. 25 FPS ruudunpäivitysnopeudella tähän kuluisi aikaa 2 sekuntia. Tämä on siis aika, jonka henkilö täytyy vähintään olla kamerakuvassa, jotta uudelleentunnistus tapahtuu. Mikäli ruudunpäivitysnopeus esimerkiksi tuplaantuu, tarvittava aika vastaavasti puolittuu. Toisaalta, mikäli piirteiden irrotusnopeutta pienennetään tai uudelleentunnistuksessa käytettävää piirrejoukon kokoa kasvatetaan, myös tarvittava aika kasvaa.

Testejä suoritettiin seuraavaksi kuudella eri tavalla. Testien tulokset löytyvät taulukoista 2–6. (1) Ensin tehtiin kontrolloitu testi samalla tavalla kuin ennen parannuksia, mutta tällä kertaa piirteet irrotettiin joka viidennellä kehyksellä ja uudelleentunnistusvertailu tehtiin, kun piirteitä oli kymmenen. Tulokset olivat tällöin hyvin selkeät. Samannäköisten henkilöiden etäisyydet olivat alle 80 ja erinäköisten henkilöiden etäisyydet olivat yli 400.

Taulukko 2. Kontrolloitu testi optimaalisissa olosuhteissa. Piirteitä ennen uudelleentunnistusta 10.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	32.2				
Musta takki ID 3	408.5	404.7			
Musta takki ID 4	411.4	407.9	35.9		
Sininen paita ID 5	62.8	74.5	403.5	404.3	

(2) Seuraavaksi ohjelmaa testattiin kontrolloimattomissa olosuhteissa. Henkilöt eivät olleet enää yhdessä kohdassa paikallaan, vaan kävelivät käytävällä. Piirteet irrotettiin edelleen joka viidennellä kehyksellä ja piirteitä kerättiin kymmenen ennen uudelleentunnistusta. Tulokset olivat huomattavasti paremmat kuin yhden piirteen avulla tehdyssä uudelleentunnistuksen testissä. Samannäköisten henkilöiden etäisyydet olivat alle 250 ja erinäköisten henkilöiden etäisyydet olivat yli 300. Samannäköiset ja erinäköiset henkilöt olivat siis tarkasti eriteltävissä nyt myös kontrolloimattomassa testissä.

Taulukko 3. Kävelytesti. Piirteitä ennen uudelleentunnistusta 10.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	153.8				
Musta takki ID 3	343.1	348.2			
Musta takki ID 4	340.7	349.1	223.6		
Sininen paita ID 5	147.1	144.1	320.3	347.5	

Seuraavat kaksi testiä tehtiin samalla tavalla kuin edellinen, mutta piirteitä kerättiin ennen uudelleentunnistusta (3) 15 ja (4) 30 kappaletta. Tulokset olivat edelleen hyviä, mutta merkittävää parannusta ei ollut havaittavissa kymmenen piirteen uudelleentunnistukseen

verrattuna. Testin kävely oli edelleen kohtalaisen kontrolloitu, joten piirrejoukon kasvatamisesta voi olla hyötyä reaali maailman ympäristössä tai useamman kameran järjestelmässä.

Taulukko 4. Kävelytesti. Piirteitä ennen uudelleentunnistusta 15.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	156.5				
Musta takki ID 3	330.4	344.1			
Musta takki ID 4	304.7	318.0	189.5		
Sininen paita ID 5	156.4	153.7	307.8	292.0	

Taulukko 5. Kävelytesti. Piirteitä ennen uudelleentunnistusta 30.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	171.7				
Musta takki ID 3	370.9	378.3			
Musta takki ID 4	360.2	366.9	198.2		
Sininen paita ID 5	184.4	183.7	352.3	353.5	

Lopuksi testattiin, miten uudelleentunnistus toimii, jos henkilö kuvataan eri kuvakulmista. Ensin henkilö käveli käytävää yhteen suuntaan, jolloin henkilö sai yhden ID:n. Hetken päästä henkilö käveli käytävää toiseen suuntaan, jolloin henkilö sai toisen ID:n ja henkilölle tehtiin uudelleentunnistus. Testi tehtiin sekä (5) 30 piirteellä, että (6) 10 piirteellä. Tulokset olivat myös tässä testissä jopa yllättävän hyvät. Samannäköisten henkilöiden etäisyydet olivat edelleen alle 250 ja erinäköisten henkilöiden etäisyydet olivat yli 300. Henkilöt olivat siis tarkasti eriteltävissä myös eri kuvakulmista kuvattuna.

Taulukko 6. Kävely eri kuvakulmista. ID:t 1, 3 ja 5 kävelevät yhteen suuntaan ja ID:t 2, 4 ja 6 kävelevät toiseen suuntaan. Piirteitä ennen uudelleentunnistusta 30.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	236.4				
Musta takki ID 3	411.2	385.3			
Musta takki ID 4	430.3	371.1	212.3		
Sininen paita ID 5	191.9	208.4	393.1	410.0	
Sininen paita ID 6	243.9	174.4	372.5	351.5	213.3

Taulukko 7. Kävely eri kuvakulmista. ID:t 1, 3 ja 5 kävelevät yhteen suuntaan ja ID:t 2, 4 ja 6 kävelevät toiseen suuntaan. Piirteitä ennen uudelleentunnistusta 10.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4	Sininen paita ID 5
Sininen paita ID 1					
Sininen paita ID 2	225.4				
Musta takki ID 3	371.8	358.8			
Musta takki ID 4	365.0	334.8	181.1		
Sininen paita ID 5	170.7	209.5	362.9	360.9	
Sininen paita ID 6	217.2	171.5	348.4	312.2	208.8

Parannetun ohjelman testitulokset olivat yllättävän hyvät, sillä uudelleentunnistus vaikuttaisi toimivan jopa erittäin hyvin testeissä, joissa testihenkilöt kävelivät käytävää pitkin.

8.3 Useamman kameran testaus

Seuraavaksi testattiin, toimiiko uudelleentunnistus myös useamman kameran välillä. Toinen kamera asennettiin eri käytävälle kuin missä ensimmäinen kamera sijaitsi. Lisäksi Yolov4-deepsort-ohjelmaa muokattiin siten, että tietokantaan tallennettiin myös tieto, mikä kamera havainnon teki. Näin henkilöt yksilöivistä tunnisteista saatiin yksilöllisiä myös kameroiden välillä.

Kahden kameran testissä saatiin parempia tuloksia kävelytestillä kuin mitä kontrollidulla testillä olemalla paikallaan. Syynä tähän voi olla kahden eri kameran erilaiset taustat, jotka paikallaan ollessa aiheuttavat havaituille henkilöille suuren etäisyyden. Kävellessä puolestaan taustat muuttuvat muutenkin, jolloin taustat eivät pääse vaikuttamaan piirteisiin niin merkittävästi.

Kahden kameran testejä tehtiin myös useita erilaisia. Kuten taulukosta 8 nähdään, kontrollitestissä, jossa henkilöt olivat paikallaan ja piirteitä kerättiin 10 ennen uudelleentunnistusta, uudelleentunnistus tunnisti kahden erinäköisen henkilön etäisyydeksi pienemmän arvon, kuin kahden samannäköisen henkilön etäisyydeksi. Tämä oli ainoa virhe, mikä testituloksissa havaittiin. Kontrollitestissä uudelleentunnistus toimi paremmin, mikäli piirteitä kerättiin suurempi joukko ennen uudelleentunnistusta.

Tulokset olivat kahden kameran testeissä selkeitä, mikäli testihenkilöt kävelivät eivätkä olleet paikallaan. Etäisyydelle pystyttiin määrittämään raja-arvo, jota suuremmat etäisyyden arvot tarkoittivat, että kyseessä on eri henkilöt ja sitä pienemmät etäisyyden arvot tarkoittivat, että kyseessä on sama henkilö. Uudelleentunnistus siis toimi myös kahden kameran välillä.

Taulukko 8. Kontrolloitu testi kahden kameran välillä. ID:t 1 ja 3 ovat yhdestä kamerasta ja ID:t 2 ja 4 ovat toisesta kamerasta. Piirteitä ennen uudelleentunnistusta 10.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4
Sininen paita ID 1				
Sininen paita ID 2	191.6			
Musta takki ID 3	401.2	431.9		
Musta takki ID 4	487.1	331.1	391.6	

Taulukko 9. Kontrolloitu testi kahden kameran välillä. ID:t 1 ja 3 ovat yhdestä kamerasta ja ID:t 2 ja 4 ovat toisesta kamerasta. Piirteitä ennen uudelleentunnistusta 30.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4
Sininen paita ID 1				
Sininen paita ID 2	147.3			
Musta takki ID 3	370.0	442.0		
Musta takki ID 4	374.6	312.6	280.4	

Taulukko 10. Kävelytesti kahden kameran välillä. ID:t 1 ja 3 ovat yhdestä kamerasta ja ID:t 2 ja 4 ovat toisesta kamerasta. Piirteitä ennen uudelleentunnistusta 30.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4
Sininen paita ID 1				
Sininen paita ID 2	254.6			
Musta takki ID 3	412.6	494.8		
Musta takki ID 4	469.1	451.7	284.9	

Taulukko 11. Kävelytesti kahden kameran välillä. ID:t 1 ja 3 ovat yhdestä kamerasta ja ID:t 2 ja 4 ovat toisesta kamerasta. Piirteitä ennen uudelleentunnistusta 10.

	Sininen paita ID 1	Sininen paita ID 2	Musta takki ID 3	Musta takki ID 4
Sininen paita ID 1				
Sininen paita ID 2	299.2			
Musta takki ID 3	412.0	435.4		
Musta takki ID 4	481.6	352.3	280.6	

8.4 Tulosten yhteenveto ja arviointi

Uudelleentunnistuksessa arviointimenetelmänä voi käyttää uudelleentunnistuksen tarkkuutta, eli miten suuressa osassa uudelleentunnistuksista uudelleentunnistus osui oikeaan. Vaihtoehtoisesti tarkkuutta on mahdollista tarkastella tasojen- n perusteella, jossa n on esimerkiksi viisi. Tämä tarkoittaa tarkkuutta, jolla uudelleentunnistettu henkilö on viiden lähimmäksi tunnistetun henkilön joukossa. Tällä tavalla uudelleentunnistuksen suorituskkyä on arvioitu esimerkiksi Campsin ja muiden tutkimuksessa (2017).

Tässä työssä tehdyt testit olivat vain kahden henkilön testejä, joten tasoihin- n perustuvaa arviointimenetelmää ei edes voida käyttää tai n on käytännössä yksi. Uudelleentunnistus siis tunnistaa henkilön joko täysin oikein tai ei. Lisäksi testit tehtiin vain selvittämään, toimiiko uudelleentunnistus vai ei. Uudelleentunnistuksia ei siis tehty suurta joukkoa vaan muutama tarkkaan tiedossa oleva vertailu. Näin menettelemällä uudelleentunnistuksen tarkkuudeksi saatiin lopulta suurimassa osassa testejä 100 %.

Testien tarkoitus siis oli todeta, toimiiko uudelleentunnistus halutulla tavalla. Lisäksi testauksen avulla saatiin konkreettinen kuva siitä, miten parametreilla voidaan vaikuttaa uudelleentunnistuksen toimintaan. Testien perusteella uudelleentunnistus toimii jopa yllättävän hyvin testiolosuhteissa.

Henkilöiden piirteitä tulisi kerätä useampia ennen uudelleentunnistuksen tekemistä. Kymmenen piirteen kerääminen ennen uudelleentunnistusta vaikuttaisi olevan riittävästi,

eikä uudelleentunnistuksen toiminnan kannalta ole suurta merkitystä, mikäli piirteitä kerätään 15 tai 30.

Testiolosuhteissa toteutettujen testien perusteella uudelleentunnistus toimii hyvin, joten uudelleentunnistusta on mahdollista lähteä kokeilemaan reaali maailman olosuhteisiin.

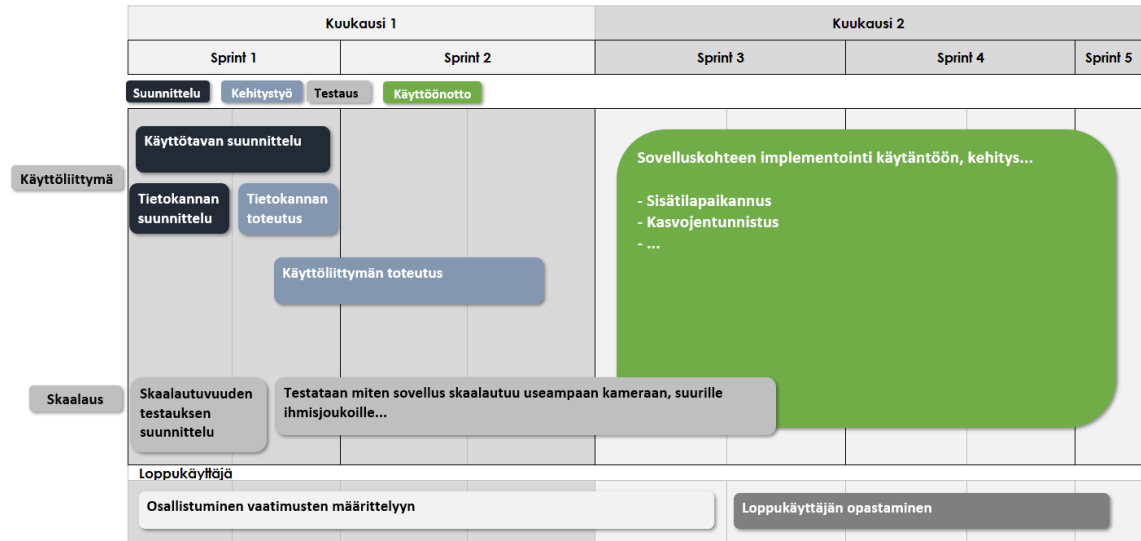
8.5 Jatkokehitys

Testit osoittivat, että uudelleentunnistus toimii, mutta kyseessä on vasta prototyyppi. Tässä vaiheessa on vielä epäselvää, miten hyvin ohjelmasta saadaan tehtyä toimiva tuote. Jatkokehityksen seuraavia askelia on kuvattu kuvassa 13 olevassa tiekartassa.

Ensimmäisenä kehityskohteena on pohtia, miten uudelleentunnistustulokset esitetään loppukäyttäjälle. Tällä hetkellä ohjelma vain tulostaa etäkonsoliin henkilöiden väliset etäisyydet sekä sen onko kyseessä ohjelman mielestä sama henkilö. Eräs mahdollinen esitystapa olisi piirtää uudelleentunnistustulokset kamerakuvaan. Tulosten esittämiseen liittyy läheisesti tietokannan suunnittelu ja miten tulokset tallennetaan tietokantaan tulosten esittämistä varten.

Toinen jatkokehitystä kaipaava asia on ohjelmiston skaalaus. Ohjelmistoa testattiin kahdella kameralla, kahdella erinäköisellä henkilöllä sekä vain muutaman minuutin ajan kerrallaan. Toistaiseksi on epävarmaa, miten ohjelma skaalautuu, kun kameroita lisätään, kamerakuvassa olevien henkilöiden määrä kasvaa ja ohjelmiston pitäisi olla päällä jatkuvasti pitkiäkin aikoja. Esimerkiksi henkilöiden piirteet ovat pelkästään uudelleentunnistusmoduulin käyttömuistissa. Toisaalta yksityisyyden kannalta on ihan hyvä, että henkilöiden piirteet nollautuisivat tietyn ajan jälkeen.

Mikäli ohjelmistoa mietitään asiakkaan näkökulmasta, kameroiden liittämisessä järjestelmään on myös kehittämistä. Tällä hetkellä jokaiselle kameralle erikseen on aukaistava verkon palomuriin portti, koska Yolov4-deepsort-ohjelmisto ottaa yhteyden kameroihin. Järkevämpi tapa olisi aukaista yksi portti Yolov4-deepsort-ohjelmiston päähän, jolloin kamerat voisivat ottaa yhteyden siihen. Toisaalta ei ole varmuutta, miten tämä saataisiin todellisuudessa toimimaan.



Kuva 13. Jatkokehityksen tiekartta.

Sovelluksen kehitys jatkuu asiakasta varten tehtävänä projektina. Kameratekniikkaa pyritään saamaan asiakkaan haluamiin tiloihin sovelluksen edellyttämällä tavalla, jonka jälkeen päästään testaamaan, miten hyvin uudelleentunnistus toimii reaaliympäristössä.

8.6 Sovelluskohteiden pohdintaa

Tässä kappaleessa pohditaan mahdollisia sovelluskohteita, joissa voitaisiin hyödyntää kehitettyä uudelleentunnistusohjelmaa sekä konenäköä laajemminkin. Kehitetty uudelleentunnistusohjelmalle yleisimpiä sovelluskohteita ovat erilaiset paikantamiseen liittyvät sovelluskohteet. Sovellus sopii tilanteisiin, joissa täytyy tietää, missä joku tietty henkilö sijaitsee.

Adaimi ja muut (2021) ovat käyttäneet uudelleentunnistusta liikennevirtojen analysointiin. Uudelleentunnistuksen ei tarvitse välttämättä kohdistua henkilöihin, vaan uudelleentunnistuksen voi tehdä myös esimerkiksi autoihin. Toki samat automallit näyttävät hyvin samalta, jolloin niitä ei pystytä yksilöimään toisistaan. Ravor & Sudarshan (2020) ovat puolestaan tutkineet uudelleentunnistuksen käyttämistä eri eläinlajien yksilöiden seurantaan.

Yksi sovelluskohde on sairaalaympäristö. Erityisesti ensiapuosaston ympäristössä tilanteet muuttuvat usein hektisiksi ja henkilövirrasta liikkuvat nopeasti huoneista toisiin.

Lisäksi muistisairaat ihmiset saattavat lähteä vaeltelemaan ja kadota. Tähän liittyen sairaalaympäristön lisäksi mahdollisia sovelluskohteita voisi olla myös hoivakotien ympäristöt. Uudelleentunnistuksen avulla voisi olla mahdollista paikantaa halutut henkilöt. Kameroiden ja uudelleentunnistuksen avulla tehtävä paikannus sopisi sairaalaympäristöön hyvin, sillä esimerkiksi signaaleihin perustuvassa paikannuksessa ongelmaksi voi muodostua signaalien aiheuttamat häiriöt terveydenhuollon laitteisiin. Lisäksi signaaliin perustuvat menetelmät vaativat, että paikannettavilla henkilöillä on mukanaan radiolähetin, kun taas kameroilla tehtävän paikannuksen avulla voidaan seurata ketä tahansa henkilöä. (Calderoni ja muut 2014)

Yhtenä sovellusympäristönä voisi olla teollisuuslaitos. Teollisuuslaitokset ovat usein laajoja alueita, joissa henkilöiden paikantamisesta voi olla hyötyä. Teollisuuslaitoksissa sovellusta olisi mahdollista käyttää kulunvalvonnan ja turvallisuusvalvonnan apuna. Sovellusta voisi soveltaa seuraamaan, ettei rajatuille alueille päädy henkilöitä, joiden ei kuuluisi alueella olla. Sovellus voisi antaa havaituista turvallisuuspoikkeamista hälytyksen.

Eräs mahdollinen sovelluskohde voisi olla myös retkeilyreiteillä tai hiihtoladuilla kadonneen henkilön paikantaminen. Tämä tosin on melko kaukaa haettu sovelluskohde, jossa ongelmaksi muodostuu esimerkiksi tarvittava kamerainfrastrukturi. Ohjelmistoa voisi myös mahdollisesti kehittää hakemaan henkilöitä, jotka omaavat tiettyjä piirteitä. Ohjelmiston avulla pystyttäisiin etsimään henkilöä, jolla on esimerkiksi tietynvärinen takki, lyhyet hiukset, ei lakkia ja olkalaukku. Tällaisen toiminnallisuuden toteuttaminen on kuitenkin kohtalaisen haastavaa. Henkilön koon puolestaan voisi saada havaittua kohtalaisen helposti. Tällöin voitaisiin hakea esimerkiksi kadonnutta pienikokoista lasta ihmisjoukosta.

Eräs sovelluskohde, jossa uudelleentunnistusta voisi käyttää hyödyksi, on erilaisten esineiden yksilöllinen tunnistaminen ja uudelleentunnistaminen. Esimerkiksi hitsaus ja polttoleikkaus jättävät yksilöllisen jäljen, jonka avulla olisi ainakin teoriassa mahdollista erottaa yksilöt toisistaan. Tätä voisi mahdollisesti käyttää hyödyksi esimerkiksi varastettujen polkupyörien tai auton katalysaattoreiden yksilöimisessä ja vertailussa. Tämäkin sovelluskohde on tosin vain ajatuksen tasolla. Ongelmana tässä sovelluskohteessa on ainakin se, että yksilöivästä esineestä pitäisi olla otettuna kuva etukäteen. Sen jälkeen esineiden joukosta olisi mahdollista etsiä tiettyä yksilöä. Esimerkiksi kamera voisi kuvata

romutettavien autojen katalysaattorit ja antaa hälytyksen, mikäli varastetuksi tiedetty katalysaattori havaitaan. Konenäköä yleisesti ilman yksilöivää uudelleentunnistusta käytetään jo tällä hetkellä esimerkiksi seulomaan valmistusvirheitä liukuhihnalta.

Uudelleentunnistus voisi toimia apuna myös myymälävarkauksien torjunnassa. Tutkimuksissa on havaittu, että isoissa vaateketjuissa myymälävarkauksiin johtavat usein tilanteet, joissa asiakkaat jäävät pyörimään liikkeen tiloihin poikkeuksellisen pitkäksi aikaa. Ongelmaa on jo lähdetty ratkaisemaan kameravalvonnan ja tekoälyn avulla. Uudelleentunnistus voisi olla yksi työkalu havaitsemaan, miten kauan asiakkaat viipyvät myymälän tiloissa. (Hedengren 2020)

Viimeisenä sovelluskohteena voisi vielä mainita liikenneturvallisuuteen liittyen itseajavat autot. Itseajavien autojen on havainnoitava ympäristöä ja erityisesti muuta liikennettä kuten jalankulkijoita. Siihen ei tosin tarvita muiden ajoneuvojen tai henkilöiden uudelleentunnistusta. Konenäkö ja objektin havaitseminen on jo tänä päivänä itseajavissa autoissa muiden tiellä liikkuvien havaitsemisen apuna.

9 EETTISTEN KYSYMYSTEN HUOMIOIMINEN KÄYTÄNNÖSSÄ

Tässä työssä kehitetty sovellus voi herättää joitain eettisyyteen liittyviä kysymyksiä. Kameravalvonta yhdistettynä tekoälyyn ja erityisesti kasvojentunnistukseen on Kiinassa laajalti käytössä ja tuntuu länsimaalaisista monesti dystooppiselta valvontayhteiskunnalta (Matikainen 2018; Valtanen 2018; Horelli 2019; Leppänen 2019). Tämän työn sovelluksessa ei ole ollenkaan kasvojentunnistusta, mutta tekoäly yhdistettynä kameravalvontaan voi siitä huolimatta herättää huolta. Kameravalvonnan ongelmallisuutta eettisestä näkökulmasta lisää se, ettei ihmisillä ole välttämättä tietoa siitä, että heistä kerätään dataa eikä tosiasiallista mahdollisuutta kieltäytyä henkilötietojen käytöstä. Valvonnan herättämään huoleen voidaan vastata mahdollisimman laajalla avoimuudella, joka yhdistetään minimalistiseen periaatteeseen.

Minimalistisella periaatteella tarkoitetaan tässä tapauksessa periaatetta, jonka mukaan yksityisyyteen liittyvää tietoa pyritään keräämään mahdollisimman vähän. Tämä voidaan ottaa huomioon jo sovellusta kehitettäessä. Esimerkiksi tämän työn uudelleentunnistushjelmistoa kehitettäessä on mahdollista tehdä valinta, että henkilöistä ei tallenneta kuvia minnekään vaan pelkästään kuvista luodut piirrevektorit. Kuvien tallentamista voidaan pitää huomattavasti suurempana loukkauksena henkilöiden yksityisyyteen, verrattuna tilanteeseen, että tallennetaan pelkästään piirteitä kuvaava lista lukuja. Lisäksi käyttökohteesta riippuen tallennetut piirrevektorit voidaan poistaa tietyin väliajoin, joka on linjassa minimalistisen periaatteen kanssa. Tällaisiin yksittäisiin valintoihin teknisillä asiantuntijoilla ja tekoälyn toteuttajilla on mahdollisuus vaikuttaa tekemisessään.

Mahdollisimman laajalla avoimuudella puolestaan voidaan kasvattaa luottamusta ja sitä kautta vaikuttaa sovelluksen käyttöönottohalukkuuteen. Kun toteutetaan mahdollisimman laajaa avoimuutta, se herättää luottamusta sekä hälventää huolia tuotteen yksityisyyteen liittyen. Mahdollisimman laaja avoimuus tarkoittaa sitä, että pyritään mahdollisimman tarkasti ja ymmärrettävällä tavalla selvittämään sovelluksen toimintaperiaate esimerkiksi niin sanotun *white paperin* avulla. White paper on helposti saatavilla oleva asiakirja, jonka voi lukea kuka tahansa ja jonka tarkoitus on nimenomaan tarjota avoimesti tietoa. Yleensä tietosuojaselosteen nimellä olevat dokumentit eivät toteuta mahdollisimman laajan avoimuuden periaatetta, vaan ovat vaikealukuista lakitekstiä, jonka tarkoituksena ei

ole sovelluksen mahdollisimman tarkan toimintaperiaatteen selvittäminen. White paperin laatijan tulisi olla tekninen asiantuntija eikä lakimies. White paperin tulisi pyrkiä vastaamaan kaikkiin mahdollisiin kysymyksiin, joita yksityisyydestä huolissaan oleva yksilö voisi haluta tietää. Sovelluksen tarkan toimintaperiaatteen lisäksi tämä tarkoittaa muun muassa tarkkaa kuvausta kerätyistä datasta ja sen elinkaaresta. Mitä tarkalleen ottaen tehdään, mitä dataa kerätään, miten dataa käsitellään, säilytetään ja hävitetään (minimalismin periaatteiden mukaan)? Esimerkki tässä työssä kehitetyn uudelleentunnistusohjelmiston white paperista löytyy liitteestä 2.

Tällaisen white paperin luettuaan käyttäjät voivat paljon luottavaisemmin mielin hyväksyä tekoälyä käytettäväksi, kun heillä on mahdollisuus saada ymmärrys siitä, miten tekoäly toimii ja miten heidän yksityisyydestään huolehditaan.

10 YHTEENVETO

Konenäkö on laaja tekoälyn osa-alue, jonka sovelluskohteet ovat hyvin monipuolisia. Koneoppiminen ja erityisesti neuroverkkoteknologiat ovat viime aikoina parantaneet erilaisia konenäkösovelluksia huomattavasti eteenpäin.

Tässä työssä tutkittiin videokuvan ja konenäön avulla tehtävää sisätilapaikannusta, henkilöiden uudelleentunnistusta sekä muita kehitettäviä sovelluskohteita konenäön avulla tehtävään videokuvan analysointiin liittyen. Kehitettävän ohjelmiston pohjalla toimi YOLOv4-deepsort-niminen objektin tunnistus- ja seurantaohjelma. Pohjalla olevan ohjelmiston päälle kehitettiin uudelleentunnistuksen toteuttava moduuli käyttäen Torchreid-ohjelmistokehikkoa. Uudelleentunnistuksen toteuttamiseen otettiin mallia Samihorm-nimisen Github-käyttäjän toteuttamasta avoimen lähdekoodin ohjelmasta.

Toteutettu ohjelmistokokonaisuus vaatii kohtalaisen paljon laskentatehoa, joten käyttötarkoitusta varten on hankittu oma Linux-palvelin, jolla ohjelmistoja ajetaan. Linux-palvelin käyttää kahta Nvidian näytönohjainta, jotka kykenevät vaatimaan rinnakkaiseen laskentaan. Linux-palvelimen ja kameroiden välisen kommunikoinnin sekä toisaalta Linux-palvelimen ja käyttäjän välisen kommunikoinnin toteuttamisessa oli omat haasteensa. Kommunikoinnit saatiin toimimaan, mutta toteutustavat eivät ole parhaat mahdolliset. Vaihtoehtoisia toteutustapoja tullaan etsimään jatkossa.

Ohjelmiston arkkitehtuurin suunnittelussa päätettiin ottaa lähestymistavaksi modulaarinen arkkitehtuuri. Ohjelmiston keskiössä on uudelleentunnistusmoduuli, johon voi liittää n-kappaletta kamera/yolov4-deepsort-moduuleita. Modulaarisuudesta on etua muun muassa ohjelmiston skaalautuvuuteen. Modulaariseen arkkitehtuuriin on helppo liittää lisää kameramoduuleita ilman muita muutoksia. Eri moduulit voivat toimia autonomisesti riippumatta muista moduuleista.

Uudelleentunnistus sisälsi kaksi pienempää ongelmaa. Tunnistettavien henkilöiden piirteet täytyi irrottaa sekä piirteitä täytyi pystyä vertailemaan toisiinsa. Näiden ongelmien ratkaisussa ratkaisevassa roolissa oli Torchreid-ohjelmistokehikko, joka on kehitetty juuri uudelleentunnistuksen tutkimusta varten. Torchreid:n avulla ohjelmistoon pystyttiin toteuttamaan piirteiden irrotus ja vertailu. Piirteiden irrotus tehtiin kamera/YOLOv4-

deepsort-moduulissa, joka tallensi henkilön piirteet sekä tunnisteiden tiedostopohjaiseen tietokantaan. Uudelleentunnistusmoduuli puolestaan lukee tietokannasta henkilöiden piirteet ja tekee piirteiden vertailun eli lopullisen uudelleentunnistuksen.

Uudelleentunnistusohjelmiston toimivuutta testattiin kontrolloiduilla testeillä. Testiympäristönä oli toimiston käytävä, jolle videokamerat asennettiin. Testeissä kamerakuvassa oli vain yksi henkilö kerrallaan. Testit suoritettiin yhdellä sekä kahdella kameralla. Uudelleentunnistusohjelmiston toimintaan pystyttiin vaikuttamaan kahdella eri parametrilla. Ensinnäkin voidaan valita, miten usein piirteet irrotetaan videokuvasta. Testien perusteella Yolov4-deepsort/kamera-moduulin suoritussyky pysyy riittävällä tasolla, mikäli piirteet irrotetaan joka viidennellä videokuvan kehyksellä. Toiseksi uudelleentunnistuksen toimivuuteen suuresti vaikuttava parametri on piirrejoukon koko ennen uudelleentunnistuksen tekemistä. Mikäli piirrejoukon koko ennen uudelleentunnistusta on vain yksi, uudelleentunnistus toimii hyvin rajoitetusti. Piirrejoukon koko eri testeissä oli 10, 15 ja 30. Kymmenellä piirteellä suoritetuissa testeissä uudelleentunnistus toimi hyvin eikä piirrejoukon kasvattamisesta 15 tai 30 piirteeseen havaittu merkittävää eroa. Piirrejoukon kasvattaminen voi kuitenkin parantaa suoritussykyä testiympäristöstä poikkeavissa reaaliympäristöissä. Testien perusteella uudelleentunnistus todettiin toimivaksi ja valmiiksi testattavaksi reaaliympäristön kohteessa. Tuote vaatii kuitenkin vielä jatkokehitystä esimerkiksi tulosten esittämisessä. Lisäksi reaaliympäristön testit tuovat luultavasti kehitettävää skaalautuvuuteen liittyen.

Työssä kehitetyn uudelleentunnistusohjelmiston avulla voidaan tunnistaa, onko havaittu henkilö havaittu kameraverkostossa jo aiemmin. Tämä mahdollistaa erilaisten sisätilapaikannussovellutusten toteuttamisen. Toki kameraverkosto voi olla myös ulkotiloissa, jolloin uudelleentunnistusta voidaan soveltaa esimerkiksi liikennevirtojen analysointiin. Muita mahdollisia sovelluskohteita on esimerkiksi myymälävarkauksien ehkäisy sekä kadonneiden henkilöiden paikantaminen kameraverkostosta.

Uudelleentunnistuksen avulla tehtävä sisätilapaikannus puuttuu jollain tasolla ihmisten yksityisyyteen ja herättää näin ollen eettisiä kysymyksiä. Näitä kysymyksiä olisi hyvä pohtia, kun sovellusta lähdetään toteuttamaan osaksi yhteiskuntaa.

11 LÄHDELUETTELO

Adaimi, G., Kreiss, S. & Alahi, A., 2021. Deep visual Re-identification with confidence. *Transportation Research Part C: Emerging Technologies*, 126.

AI Optio, 2018. Kuinka neuroverkko toimii? [Verkkodokumentti] Youtube. Katsottavissa: <https://www.youtube.com/watch?v=APzICkVo2Q0> [Viitattu 17.5.2021]

Bochkovskiy, A., Wang, C-Y. & Liao, H-Y. M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. Saatavilla: <https://arxiv.org/pdf/2004.10934v1.pdf> [Viitattu 28.5.2021].

Brunetti, A., Buongiorno, D., Trotta, G. F., Bevilacqua, V., 2018. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300, s. 17-33.

Camps, O., Gou, M., Hebble, T., Karanam, S., Lehmann, O., Li, Y., Radke, R. J., Wu, Z., Xiong, F., 2017. From the Lab to the Real World: Re-identification in an Airport Camera Network. *Transactions on Circuits and Systems for Video Technology*, 27, s. 540-553.

Calderoni, L., Ferrara, M., Franco, A. & Maio, D., 2014. Indoor localization in a hospital environment using Random Forest classifiers. *Expert Systems with Applications*, 42, s. 125-134.

Chen, A. T-Y., Biglari-Abhari, M., Wang, K. I-K., 2019. Investigating fast re-identification for multi-camera indoor person tracking. *Computers and Electrical Engineering*, 77, s. 273-288.

COCO Consortium, 2020. COCO 2020 Panoptic Segmentation Task. Tasks, Panoptic 2020. [Verkkodokumentti] Saatavissa: <https://cocodataset.org/#panoptic-2020> [Viitattu 2.9.2021].

COCO Consortium, 2017. COCO 2017 training dataset. Saatavissa: <https://cocodataset.org/#explore?id=386298> [Viitattu 2.9.2021].

Feng, X., Jiang, Y., Yang, X., Du, M., Li, X., 2019. Computer vision algorithms and hardware implementations: A survey. *Integration, the VLSI Journal*, 69, s. 309-320.

Haaranen, M. & Allonen, J., 2020. Kameravalvonnassa käytettävän tekoälyn hyödyntäminen Suomessa. Jyväskylän yliopisto, Informaatioteknologian tiedekunta. Saatavissa: <http://urn.fi/URN:NBN:fi:juu-202005253424> [Viitattu 19.8.2021].

Hedengren, 2020. Tekoäly tulee kameravalvontaan – mitä hyötyjä se tuo ja kenelle? [Verkkootikeli]. Saatavissa: <https://www.hedengren.com/fi/tekoaly-tulee-kameravalvontaan-mita-hyotyja-se-tuo-ja-kenelle> [Viitattu 19.8.2021].

Horelli, M., 2019. Kiinassa metrolipun ja pikaruoan voi jo maksaa vilauttamalla kasvoja – Myös lännessä viranomaiset innostuivat kasvojentunnistuksesta. [Verkkouutinen] Suomen Kuvalehti, Otavamedia. Saatavissa: <https://suomenkuvalehti.fi/jutut/ulkomaat/kiinassa-metrolipun-ja-pikaruoan-voi-jo-maksaa-vilauttamalla-kasvoja-myos-lannessa-viranomaiset-innostuivat-kasvojentunnistuksesta/> [Viitattu 17.8.2021].

Hyyti, H., 2016. Automaatio-ja systeemitekniikan laboratoriotyöt, Konenäkö robotin ohjauksessa. [Verkkodokumentti] Espoo: Aalto yliopisto. Saatavissa: <https://mycourses.aalto.fi/course/view.php?id=5074§ion=1> [Viitattu 20.4.2021].

IGI Global, 2021. What is Re-Identification. Dictionary, Re-Identification. Saatavissa: <https://www.igi-global.com/dictionary/re-identification/24583> [Viitattu 23.8.2021].

Jiang, W., & Yin, Z., 2017. Combining passive visual cameras and active IMU sensors for persistent pedestrian tracking. *Journal of Visual Communication and Image Representation*, 48, s. 419-431.

Jurić, D., 2015. Object Tracking: Kalman Filter with Ease. [Verkkodokumentti]. Saatavissa: <https://www.codeproject.com/articles/865935/object-tracking-kalman-filter-with-ease> [Viitattu 12.5.2021].

Kawaji, H., Hatada, K., Yamasaki, T. & Aizawa, K., 2010. Image-based Indoor Positioning System: Fast Image Matching using Omnidirectional Panoramic Images. University of Tokyo.

Keim, R., 2020. How Many Hidden Layers and Hidden Nodes Does a Neural Network Need? [Verkkodokumentti]. Saatavissa: <https://www.allaboutcircuits.com/technical-articles/how-many-hidden-layers-and-hidden-nodes-does-a-neural-network-need/> [Viitattu 18.5.2021].

Leppänen, M., 2019. San Francisco kieltää kasvot tunnistavan teknologian käytön poliisilta ja hallituksen virastoilta. [Verkkouutinen] STT, Yleisradio. Saatavissa: <https://yle.fi/uutiset/3-10783978> [Viitattu 17.8.2021].

Li, Y., Liu, L., Zhu, L. & Zhang, H., 2021. Person re-identification based on multi-scale feature learning. Knowledge-Based Systems, 228.

Lin, T-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L. & Dollár, P., 2015. Microsoft COCO: Common Objects in Context. [Verkkodokumentti]. Saatavissa: <https://arxiv.org/abs/1405.0312> [Viitattu 9.6.2021].

Liu, G. & Wu, J., 2021. Unsupervised person re-identification by Intra-Inter Camera Affinity Domain Adaptation. Journal of Visual Communication and Image Representation.

Matikainen, J., 2018. Entä jos jokainen tekosi tallentuisi kameralle ja sinut pisteytettäisiin kansalaisena? Kiinassa se on pian totta. [Verkkouutinen] Yleisradio. Saatavissa: <https://yle.fi/uutiset/3-10135093> [Viitattu 17.8.2021].

Mautz, R., 2012. Indoor positioning technologies. ETH Zürich Habilitation Thesis. Saatavissa: <https://doi.org/10.3929/ethz-a-007313554> [Viitattu 3.6.2021].

Nguyen, D.T, Li, W. & Ogunbona, O., 2016. Human detection from images and videos: A survey. Pattern Recognition, 51, s. 147-175.

OpenCV team, 2021. OpenCV, About. [Verkkodokumentti]. Saatavissa: <https://opencv.org/about/> [Viitattu 28.5.2021].

Pham, T.T.T., Le, T.-L., Vu, H., Dao, T. K. & Nguyen, V. T., 2017. Fully-automated person re-identification in multi-camera surveillance system with a robust kernel descriptor and effective shadow removal method. Image and Vision Computing, 59, s. 44-62.

Pixabay 2021. Chess pawn. [Verkkosivusto] Saatavissa: <https://pixabay.com/photos/chess-pawn-game-plan-queen-mate-3203086/> [Viitattu 2.9.2021].

Ravoor, P.C. & Sudarshan, T.S.B., 2020. Deep Learning Methods for Multi-Species Animal Re-identification and Tracking – a Survey. Computer Science Review, 38.

Russel, S. & Norvig, P., 2016. Artificial Intelligence - A Modern Approach. 3 painos. Harlow (England): Pearson Education Limited, 1132 s. ISBN-10: 1-292-15396-2, ISBN-13: 978-1-292-15396-4

Sachdev, H., 2020. Choosing number of Hidden Layers and number of hidden neurons in Neural Networks [Verkkodokumentti]. Saatavissa: <https://www.linkedin.com/pulse/choosing-number-hidden-layers-neurons-neural-networks-sachdev> [Viitattu 18.5.2021].

Samihorm, 2021. Multi-Camera-Person-Tracking-and-Re-Identification. Github. Saatavissa: <https://github.com/samihormi/Multi-Camera-Person-Tracking-and-Re-Identification> [Viitattu 26.7.2021].

Shapiro, L. G. & Stockman, G. C. 2001. Computer vision. Upper SaddleRiver (N.J.): Prentice Hall, 580 s. ISBN 0-13-030796-3

Sichkar, V., 2020. Introduction into YOLO v3. [Verkkodokumentti] Youtube. Katsottavissa: <https://www.youtube.com/watch?v=vRqSO6RsptU> [Viitattu 3.6.2021]

Stahl, B. C., 2021. Artificial Intelligence for a Better Future. An Ecosystem Perspective on the Ethics of AI and Emerging Digital Technologies. De Montfort University. Leicester, UK.. Saatavissa: <https://doi.org/10.1007/978-3-030-69978-9> [Viitattu 10.8.2021].

Tay, C.-P. & Yap, K.-H., 2021. Attribute saliency network for person re-identification. Image and Vision Computing.

Tekoälyä käsittelevä korkean tason asiantuntijaryhmä, 2019. Luotettavaa tekoälyä koskevat eettiset ohjeet [Verkkodokumentti]. Euroopan komissio. Saatavissa: <https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1/language-fi/format-PDF> [Viitattu 18.8.2021].

Tekoäly.info, 2021. Tekoälyn historia. Vaasa: Skycode Oy Saatavissa: https://tekoaly.info/tekoaly_historia/ [Viitattu 18.3.2021].

The AI Guy, 2021. yolov4-deepsort. Github. Saatavissa: <https://github.com/theAIGuysCode/yolov4-deepsort> [Viitattu 25.5.2021].

Työ- ja elinkeinoministeriö, 2019. Edelläkävijänä tekoälyaikaan. Tekoälyohjelman loppuraportti. Työ- ja elinkeinoministeriön julkaisuja, 2019:23.

Valtanen, M., 2018. Mihin uskot, missä liikut ja kenen kanssa? – Valtion ”Tarkat silmät” valvovat pian jokaista kiinalaista. [Verkkouutinen] Suomen Kuvalehti, Otavamedia. Saatavissa: <https://suomenkuvalehti.fi/jutut/ulkomaat/aasia-tyynimeri/mihin-uskot-missa-liikut-ja-kenen-kanssa-valtion-tarkat-silmat-valvovat-pian-jokaista-kiinalaista/> [Viitattu 17.8.2021].

Wei, Y. & Akinci, B., 2019. A vision and learning-based indoor localization and semantic mapping framework for facility operations and management. Automation in Construction, 107. Saatavissa: <https://doi.org/10.1016/j.autcon.2019.102915> [Viitattu 4.6.2021].

Werner, M. & Kessel, M., 2011. Indoor Positioning Using Smartphone Camera. Munich, Germany: Ludwig-Maximilians-University.

Zheng, L., Yang, Y. & Hauptmann, A.G., 2016. Person Re-identification: Past, Present and Future. University of Technology at Sydney, NSW, Australia.

Zhou, K. & Xiang, T., 2019. Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. University of Surrey, UK. Saatavissa: <https://arxiv.org/abs/1910.10093> [Viitattu 26.7.2021].

Liite 1. Uudelleentunnistusohjelmiston white paper.

- Kamera lähettää jatkuvaa videokuvaa laskentapalvelimelle.
- Laskentapalvelimella pyörii ohjelma, joka havaitsee videokuvasta henkilöitä ja “merkitsee” henkilöt ympäröivillä laatikoilla.
- Henkilön havaittuaan ohjelma irrottaa laatikon pikselit ja syöttää pikselidatan neuroverkolle, joka palauttaa henkilöä kuvaavan piirrejoukon. Tämä piirrejoukko on käytännössä neuroverkon tuottama 2048-kokoinen lista lukuja. Esimerkiksi [0.123, 0.456, 0.789... 0.321].
- Tämä lukulista kuvaa henkilön piirteitä ja sitä käytetään vertailemaan eri henkilöiden ulkonäön samankaltaisuutta.
- Takaisinmallintaminen piirteistä kuvaksi ei ole mahdollista tai se vaatisi huomattavaa jatkokehitystä.
- Näitä piirteitä muodostetaan henkilöstä joka viidennellä videokuvan kehyksellä.
- Piirteet siirretään väliaikaisen tietokannan avulla toisen ohjelman käyttöön, joka toteuttaa piirteiden avulla henkilöiden uudelleentunnistuksen.
- Ohjelma siis vertailee, onko havaittu henkilö samannäköinen, kuin joku aiemmin havaittu henkilö.
- Väliaikainen tietokanta ja kaikki piirteet pyyhitään muistista x väliajoin.